APPROACHES FOR ENRICHING AND IMPROVING TEXTUAL KNOWLEDGE BASES

Von der Fakultät für Elektrotechnik und Informatik
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des akademischen Grades

DOKTOR DER NATURWISSENSCHAFTEN

Dr. rer. nat.

genehmigte Dissertation
von

M. Sc. Besnik Fetahu

geboren am 21. September 1986, in Prishtina, Kosovo

Hannover, Deutschland, 2017

Referent: Prof. Dr. techn. Wolfgang Nejdl
Korreferent: Prof. Dr. Maarten de Rijke
Korreferent: Prof. Dr.-Ing. Bodo Rosenhahn
Tag der Promotion: 08.11.2017

ABSTRACT

Verifiability is one of the core editing principles in Wikipedia, where editors are encouraged to provide *citations* for the added statements. Statements can be any arbitrary piece of text, ranging from a sentence up to a paragraph. However, in many cases, citations are either outdated, missing, or link to non-existing references (e.g. dead URL, moved content etc.). In total, 20% of the cases such citations refer to *news* articles and represent the second most cited source. Even in cases where citations are provided, there are no explicit indicators for the span of a citation for a given piece of text. In addition to issues related with the verifiability principle, many Wikipedia entity pages are incomplete, with relevant information that is already available in online news sources missing. Even for the already existing citations, there is often a delay between the news publication time and the reference time.

In this thesis, we address the aforementioned issues and propose automated approaches that enforce the *verifiability* principle in Wikipedia, and suggest relevant and missing news references for further enriching Wikipedia entity pages. To this end we make the following contributions as part of this thesis:

- *Citation recommendation* – we address the problem of finding and updating news citations for statements in Wikipedia entity pages. We propose a two-stage approach for this problem. First, we classify each statement whether it requires a news citation or citations from other categories (e.g. web, book, journal, etc.). Second, for statements that require a news citation, we formalize three properties of what makes a good citation, namely: (i) the citation should entail the Wikipedia statement, (ii) the statement should be central to the citation, and (iii) the citation should be from an authoritative source. We combine standard information retrieval techniques, where we use the statement to query a news collection, and build classification models based on the three properties to determine the most appropriate citation.
- *Citation span* – from the already existing citations in Wikipedia entity pages and the ones we recommend in our first problem, we propose an automated approach which determines the span of such citations. We approach this problem by classifying which textual fragments in a paragraph are covered or hold true given a citation. We propose a sequence classification approach where for a paragraph and a citation, we determine the citation span at a fine-grained level.
- *News suggestion* – to account for the ever evolving nature of Wikipedia entities, with relevant information published on a daily basis in news articles, we propose a two-stage supervised approach for this problem. First, we suggest news articles to Wikipedia entities (article-entity placement) relying on a rich set of features which take into account the *salience* and *relative authority* of entities, and the *novelty* of news articles to entity pages. Second, we determine the exact section in the entity page for the input article (article-section placement) guided by class-based section templates.

We perform extensive evaluation with real-world datasets, on news collections with more than 20 million news articles, and on the entire set of english Wikipedia entity pages. Our approaches perform with high accuracy on the three problems we address and show superior performance when compared to existing baselines and state of the art approaches.

Keywords: *citation recommendation, citation span, news suggestion, wikipedia enrichment*

ZUSAMMENFASSUNG

Nachweisbarkeit ist eines der zentralen Editierungs-Prinzipien in Wikipedia. Editoren werden dazu angehalten, ihre hinzugefügten Aussagen mittels *Zitierungen* zu belegen. Aussagen können dabei beliebige Textstücke sein, von Sätzen bis hin zu einem Absatz. In vielen Fällen sind Zitierungen jedoch veraltet, fehlen oder verweisen auf nicht existierende Referenzen (z.B. tote URLs oder verschobene Inhalte). In 20% der Fälle verweisen Zitierungen auf *News*-Artikel, die zweithäufigste Art der Zitierung in Wikipedia. Selbst in den Fällen, in denen Zitierungen vorhanden sind, fehlen Angaben über die Spanne des Textes, die durch die Zitierung abgedeckt wird. Unabhängig von Problemen in Zusammenhang mit dem Nachweisbarkeits-Prinzip sind viele Wikipedia-Artikel unvollständig, wobei oft relevante Information fehlt, die bereits in Online News-Quellen verfügbar ist. Auch für die bereits hinzugefügten Zitierungen existiert oft eine Verzögerung zwischen der Veröffentlichungszeit des News-Artikels und der Zeit, zu der der Artikel in Wikipedia referenziert wurde.

In dieser Arbeit beschäftigen wir uns mit den aufgezeigten Problemen und stellen automatisierte Ansätze vor, die das *Nachweisbarkeits*-Prinzip in Wikipedia durchsetzen und relevante und fehlende News-Referenzen vorschlagen mit dem Ziel, die Qualität von Wikipedia-Artikeln zu erhöhen. Diese Arbeit enthält die folgenden Beiträge:

- *Zitierungsempfehlungen* – Wir beschäftigen uns mit dem Problem des Findens und Erneuerns von News-Zitierungen für Aussagen in Wikipedia-Artikeln. Für dieses Problem stellen wir einen zweiteiligen Ansatz vor. Zunächst bestimmen wir mittels Klassifizierung, ob eine Aussage eine News-Zitierung oder eine Zitierung aus einer anderen Kategorie (z.B. Web, Bücher, Zeitschriften) benötigt. Im zweiten Schritt bestimmen wir drei Eigenschaften, die eine gute Zitierung ausmachen: (i) die Zitierung sollte die Wikipedia-Aussage enthalten, (ii) die Aussage sollte eine zentrale Rolle im zitierten Artikel einnehmen und (iii) der zitierte Artikel sollte aus einer verlässlichen Quelle stammen. Wir kombinieren Standardtechniken des Information Retrieval und verwenden die gegebene Aussage, um eine passende News-Sammlung zusammenzustellen. Weiterhin entwickeln wir Klassifizierungsmodelle basierend auf den drei genannten Eigenschaften, um die passendste Zitierung zu ermitteln.
- *Zitierungsspanne* – Aus den bereits existierenden Zitierungen in Wikipedia Artikeln und den Zitierungen, die wir in unserem ersten Problem vorschlagen, entwickeln wir einen automatisierten Ansatz zur Bestimmung der Spanne einer Zitierung. Dazu klassifizieren wir, welche Textbausteine eines Absatzes durch eine Zitierung abgedeckt werden, bzw. als wahr angesehen werden können. Wir stellen einen Sequenzklassifizierungsansatz vor, der die Spanne einer Zitierung bezogen auf einen Absatz im Detail bestimmen kann.
- *News-Vorschläge* – Unter Berücksichtigung der ständigen Veränderung von Wikipedia-Artikeln sowie der Tatsache, dass täglich neue relevante Informationen in Online News-Artikeln veröffentlicht werden, stellen wir einen zweiteiligen überwachten Ansatz vor. Zunächst schlagen wir News-Artikel für Wikipedia-Artikel vor (article-entity placement), basierend auf einer ergiebigen Menge an Features, die die *relative Bedeutung* eines Wikipedia-Artikels und die *Neuheit* von News-Artikeln berücksichtigen. Im zweiten Schritt bestimmten wir die Sektion des Wikipedia-Artikels, für die der News-Artikel vorgeschlagen werden soll (article-section placement).

Wir führen umfassende Evaluationen mit real-world Datensätzen von News-Sammlungen mit mehr als 20 Millionen News-Artikeln und der gesammten Menge englischer Wikipedia-Artikel durch. Unsere Ansätze erzielen hohe Genauigkeit und übertreffen die Leistung von existierenden Baselines und State-of-the-Art Ansätzen.

Schlagwörter: *Zitierungsempfehlungen, Zitierungsspanne, News-Vorschläge, Wikipedia Anreicherung*

ACKNOWLEDGMENTS

I would like to acknowledge and thank everyone who have supported and helped me shape as a researcher and person throughout the course of my doctoral studies. First and foremost, I would like to thank my advisor Prof. Dr. techn. Wolfgang Nejdl. He provided the perfect environment and guidance throughout these years, helping me shape as a researcher and successfully conduct the work published in this thesis. He was and still remains a role model and inspiration for my future career and path.

Special thanks to Prof. Avishek Anand and Prof. Katja Markert, for their close collaboration and the countless discussions, which helped me learn and develop as a researcher. I am also very grateful to Dr. Stefan Dietze for his guidance and introducing me to many exciting topics, projects, and providing helpful feedback and discussions for the many research ideas we developed together.

I thank all the colleagues at L3S, with many of whom we became thick friends through the course of our PhD studies. With Ujwal, Ricardo, Kaweh, Patrick, Bernardo, and many others, we shared lots of moments, countless discussions, sleepless night catching the deadlines, and learned a lot from each other. Thank you.

I would like to thank a good old friend, Flakron, who was always there to celebrate and discuss all the successes and setbacks through this PhD, and being a great help with his suggestions about all the new Web technologies.

I thank Mary, for her unconditional support and being there for me during the most important part of my PhD. She was the safe haven and the escape from the hectic of countless experiments, late working hours that came along with the PhD. Thank you, I cherish your caring, help and understanding through this phase of my life.

I would like to thank my family for being there for me, for their support, love, caring and understanding. Through the course of my PhD, especially my mother, they came to learn the different paper submission time-zones, and conference acronyms. This was all possible because of you, and I dedicate this to you all.

FOREWORD

The work in this thesis and generally throughout the entire course of the Ph.D studies I have published and co-authored several papers and journals in the fields of Text Mining, Natural Language Process, Semantic Web, Information Retrieval and Human-Computer Interaction.

The core contributions of this thesis in the individual chapters are published in the following venues:

1. The contributions in Chapter 5, which deal with the problem of news suggestion for Wikipedia articles are published in:

   - [FMNA16] Besnik Fetahu, Katja Markert, Wolfgang Nejdl, Avishek Anand: Finding News Citations for Wikipedia. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016, pages 337–346.

2. The contributions in Chapter 6, which deal with the problem of determining the citation span for references in Wikipedia are published in:

   - Besnik Fetahu, Katja Markert, Avishek Anand: Fine Grained Citation Span for References in Wikipedia. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP 2017, Copenhagen, Denmark, September 7-11, 2017 (to appear).

3. The contributions in Chapter 7, which deals with the problem of automatically suggesting news articles to Wikipedia is published in:

   - [FMA15] Besnik Fetahu, Katja Markert, Avishek Anand: Automated News Suggestions for Populating Wikipedia Entity Pages. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM 2015, Melbourne, Australia, October 19 - 23, 2015, pages 323–332.

4. Finally, in Chapter 3 we present a study of the lag between Wikipedia and news, and in Chapter 8 we show an application use case of entity search in structured datasets, which are published in:

- [FAA15] Besnik Fetahu, Abhijit Anand, and Avishek Anand. How much is wikipedia lagging behind news. In *Proceedings of the ACM Web Science Conference, WebSci 2015*, Oxford, United Kingdom, June 28 - July 1, 2015, pages 28:1–28:9.
- [FGD15] Besnik Fetahu, Ujwal Gadiraju, and Stefan Dietze. Improving Entity Retrieval on Structured Data. In *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference*, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I, pages 474–491.

The complete list of publications during my PhD is shown below:

1. Besnik Fetahu, Katja Markert, Avishek Anand: Fine Grained Citation Span for References in Wikipedia. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP 2017, Copenhagen, Denmark, September 7-11, 2017 (to appear).

2. Ran Yu, Ujwal Gadiraju, Besnik Fetahu, and Stefan Dietze: FuseM: Query-Centric Data Fusion on Structured Web Markup. In *Proceedings of the 33rd IEEE International Conference on Data Engineering, ICDE*, San Diego, CA, USA, April 19-22, 2017, pages 179–182.

3. Ujwal Gadiraju, Besnik Fetahu, Rickardo Kawase, Patrick Siehndel, and Stefan Dietze: Is the Crowd Smarter Than a 5th Grader? Using Worker Self-Assessments for Competence-based Pre-Selection. In *ACM Transactions of Computer-Human Interaction*, 2017 (to appear).

4. Besnik Fetahu, Katja Markert, Wolfgang Nejdl, Avishek Anand: Finding News Citations for Wikipedia. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, pages 337–346.

5. Ran Yu, Ujwal Gadiraju, Xiaofei Zhu, Besnik Fetahu, Stefan Dietze: Towards Entity Summarisation on Structured Web Markup. In *Proceedings of the Semantic Web - ESWC 2016 Satellite Events*, Heraklion, Crete, Greece, May 29 - June 2, 2016, Revised Selected Papers, pages 69–73.

6. Ran Yu, Besnik Fetahu, Ujwal Gadiraju, Stefan Dietze: A Survey on Challenges in Web Markup Data for Entity Retrieval. In *Proceedings of the ISWC 2016 Posters & Demonstrations Track co-located with 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 19, 2016.*

7. Davide Taibi, Giovanni Fulantelli, Stefan Dietze, Besnik Fetahu: Educational Linked Data on the Web - Exploring and Analysing the Scope and Coverage. In *Open Data for Education - Linked, Shared, and Reusable Data for Teaching and Learning*, pages 16–37.

8. Jakob Beetz, Ina Blümel, Stefan Dietze, Besnik Fetahu, Ujwal Gadiraju, Martin Hecher, Thomas Krijnen, Michelle Lindlar, Martin Tamke, Raoul Wessel, Ran Yu: In *3D Research Challenges in Cultural Heritage II - How to Manage Data and Knowledge Related to Interpretative Digital 3D Reconstructions of Cultural Heritage*, pages 231–255.

9. Davide Taibi, Saniya Chawla, Stefan Dietze, Ivana Marenzi, Besnik Fetahu: Exploring TED talks as linked data for education. BJET 46(5): 1092-1096 (2015)

10. Besnik Fetahu, Katja Markert, Avishek Anand: Automated News Suggestions for Populating Wikipedia Entity Pages. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015*, Melbourne, VIC, Australia, October 19 - 23, 2015, pages 323–332.

11. Ujwal Gadiraju, Besnik Fetahu, Ricardo Kawase: Training Workers for Improving Performance in Crowdsourcing Microtasks. In *Design for Teaching and Learning in a Networked World - 10th European Conference on Technology Enhanced Learning, EC-TEL 2015*, Toledo, Spain, September 15-18, 2015, Proceedings, pages 100–114.

12. Ujwal Gadiraju, Patrick Siehndel, Besnik Fetahu, Ricardo Kawase: Breaking Bad: Understanding Behavior of Crowd Workers in Categorization Microtasks. In *Proceedings of the 26th ACM Conference on Hypertext & Social Media, HT 2015*, Guzelyurt, TRNC, Cyprus, September 1-4, 2015, pages 33–38.

13. Besnik Fetahu, Ujwal Gadiraju, Stefan Dietze: Improving Entity Retrieval on Structured Data. In *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference*, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I, pages 474–491.

14. Besnik Fetahu, Abhijit Anand, Avishek Anand: How much is Wikipedia Lagging Behind News? In *Proceedings of the ACM Web Science Conference, WebSci 2015*, Oxford, United Kingdom, June 28 - July 1, 2015, pages 28:1–28:9.

15. Ran Yu, Ujwal Gadiraju, Besnik Fetahu, Stefan Dietze: Adaptive Focused Crawling of Linked Data. In *Web Information Systems Engineer-*

ing - WISE 2015 - 16th International Conference, Miami, FL, USA, November 1-3, 2015, Proceedings, Part I, pages 554–569.

16. Davide Taibi, Giovanni Fulantelli, Stefan Dietze, Besnik Fetahu: Towards Analysing the Scope and Coverage of Educational Linked Data on the Web. In *Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015*, Florence, Italy, May 18-22, 2015 - Companion Volume, pages 705–710.

17. Besnik Fetahu, Stefan Dietze, Bernardo Pereira Nunes, Marco Antonio Casanova, Davide Taibi, Wolfgang Nejdl: A Scalable Approach for Efficiently Generating Structured Dataset Topic Profiles. In *The Semantic Web: Trends and Challenges - 11th International Conference, ESWC 2014*, Anissaras, Crete, Greece, May 25-29, 2014. Proceedings, pages 519–534.

18. Bernardo Pereira Nunes, Alexander Arturo Mera Caraballo, Ricardo Kawase, Besnik Fetahu, Marco A. Casanova, Gilda Helena Bernardino de Campos: A Topic Extraction Process for Online Forums. In *IEEE 14th International Conference on Advanced Learning Technologies, ICALT 2014*, Athens, Greece, July 7-10, 2014, pages 541–543.

19. Davide Taibi, Stefan Dietze, Besnik Fetahu, Giovanni Fulantelli: Exploring type-specific topic profiles of datasets: a demo for educational linked data. In *Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference, ISWC 2014*, Riva del Garda, Italy, October 21, 2014., pages 353–356.

20. Besnik Fetahu, Ujwal Gadiraju, Stefan Dietze: Crawl Me Maybe: Iterative Linked Dataset Preservation. In *Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference, ISWC* 2014, Riva del Garda, Italy, October 21, 2014., pages 433–436.

21. Bernardo Pereira Nunes, Ricardo Kawase, Besnik Fetahu, Marco A. Casanova, Gilda Helena Bernardino de Campos: Educational Forums at a Glance: Topic Extraction and Selection. In *Web Information Systems Engineering - WISE 2014 - 15th International Conference*, Thessaloniki, Greece, October 12-14, 2014, Proceedings, Part II, pages 351–364.

22. Besnik Fetahu, Stefan Dietze, Bernardo Pereira Nunes, Marco Antonio Casanova, Davide Taibi, Wolfgang Nejdl: What's all the data about?: creating structured profiles of linked data on the web. In *23rd International World Wide Web Conference, WWW '14*, Seoul, Republic of Korea, April 7-11, 2014, Companion Volume, pages 261–262.

23. Bernardo Pereira Nunes, Alexander Arturo Mera Caraballo, Marco Antonio Casanova, Besnik Fetahu, Luiz André P. Paes Leme, Stefan Dietze: Complex Matching of RDF Datatype Properties. In *Database and Expert Systems Applications - 24th International Conference, DEXA 2013*, Prague, Czech Republic, August 26-29, 2013. Proceedings, Part I, pages 195–208.

24. Davide Taibi, Giovanni Fulantelli, Stefan Dietze, Besnik Fetahu: Evaluating Relevance of Educational Resources of Social and Semantic Web. In *Scaling up Learning for Sustained Impact - 8th European Conference, on Technology Enhanced Learning, EC-TEL 2013*, Paphos, Cyprus, September 17-21, 2013. Proceedings, pages 637–638.

25. Bernardo Pereira Nunes, Stefan Dietze, Marco Antonio Casanova, Ricardo Kawase, Besnik Fetahu, Wolfgang Nejdl: Combining a Co-occurrence-Based and a Semantic Measure for Entity Linking. In *The Semantic Web: Semantics and Big Data, 10th International Conference, ESWC 2013*, Montpellier, France, May 26-30, 2013. Proceedings, pages 548–562.

26. Besnik Fetahu, Bernardo Pereira Nunes, Stefan Dietze: Summaries on the Fly: Query-Based Extraction of Structured Knowledge from Web Documents. In *Web Engineering - 13th International Conference, ICWE 2013*, Aalborg, Denmark, July 8-12, 2013. Proceedings, pages 249–264.

27. Bernardo Pereira Nunes, Ricardo Kawase, Besnik Fetahu, Stefan Dietze, Marco A. Casanova, Diana Maynard: Interlinking Documents based on Semantic Graphs. In *17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems, KES 2013*, Kitakyushu, Japan, 9-11 September 2013, pages 231–240.

28. Bernardo Pereira Nunes, Besnik Fetahu, Marco Antonio Casanova: Cite4Me: Semantic Retrieval and Analysis of Scientific Publications. In *Proceedings of the LAK Data Challenge*, Leuven, Belgium, April 9, 2013.

29. Bernardo Pereira Nunes, Besnik Fetahu, Stefan Dietze, Marco A. Casanova: Cite4Me: A Semantic Search and Retrieval Web Application for Scientific Publications. In *Proceedings of the ISWC 2013 Posters & Demonstrations Track*, Sydney, Australia, October 23, 2013, pages 25–28.

30. Besnik Fetahu, Stefan Dietze, Bernardo Pereira Nunes, Davide Taibi, Marco Antonio Casanova: Generating structured Profiles of Linked Data Graphs. In *Proceedings of the ISWC 2013 Posters & Demonstrations Track*, Sydney, Australia, October 23, 2013, pages 113–116

31. Besnik Fetahu, Bernardo Pereira Nunes, Stefan Dietze: Towards focused knowledge extraction: query-based extraction of structured summaries. WWW (Companion Volume) 2013: 77-78

32. Davide Taibi, Besnik Fetahu, Stefan Dietze: Towards integration of web data into a coherent educational data graph. In *22nd International World Wide Web Conference, WWW '13*, Rio de Janeiro, Brazil, May 13-17, 2013, Companion Volume, pages 419–424.

33. Besnik Fetahu, Ralf Schenkel: Retrieval evaluation on focused tasks. In *The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12*, Portland, OR, USA, August 12-16, 2012, pages 1135–1136.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

The advent of Internet and Web 2.0 platforms, has led to major societal shifts. Nowadays, Web users have the means to consume and create content as part of various sharing platforms like Social Media, Blogs and other platforms. The increasing number of users in the Web [Sta] has shifted the focus of many organizations towards providing content online. This has led to many successful digitization projects like the Internet Archive [Arc], the Million Books Project [Pro], The New York Times [San08] etc. The impact of such digitization has societal benefits, in which information is easier accessible thus leading to a more informed society. For instance, in US alone, nearly 40% of users consume their daily news through online news media platforms [Cen].

Apart from organizations that follow a certain discourse on providing information like news media, there is the other spectrum of the Web, where information is a direct result of the interaction between users and Web applications. The engagement and the number of users directly correlates to the value of an application. Examples include Social Media platforms like Twitter, Facebook etc., where user engagement is of importance to the success of these applications [Rib14]. Such a phenomena is described by Simon [Sim71], where user attention and engagement are some of the key available resources for organizations. This similarly applies to Web applications.

In this respect, one of the most known examples of such synergies between Web application and users on the Web is Wikipedia [Enc]. It represents an open, collaborative effort of creating encyclopedic content by Web users. At its core are Wikipedia editors, who provide content according to established principles and editing policies [pol]. There are approximately 284 different language versions of Wikipedia [oW]. Only in the english Wikipedia there are roughly 5 million articles, and a total of 30 million registered editors across all localized Wikipedias. The dynamics of content creation in Wikipedia, and the organizational structure and collaboration between editors has been subject to extensive research [KGC12, KPSM07, ZPL12, PHT09, WRTH15].

Wikipedia, is one of the top visited websites overall[1]. Due to the large number of editors, and its openness in terms of added content, to provide quality assurances, there are guidelines and policies [pol], editor categorizations (e.g. *admins* or *novice* editors), and finally each revision of an article can be edited or deleted by other peer-editors. Despite the fact that the nature of these policies are guidelines and are not enforced, studies [MOM+15] show that Wikipedia in specific domains achieves comparable quality to expert curated encyclopedia like Britannica [Bri].

The value of Wikipedia has been widely acknowledged. It serves as the backbone for a wide range of applications. It is used to construct knowledge graphs like DBpedia [BLK+09], YAGO [SKW07], which are included in major search engines like Google KnowledgeGraph or Apple's Siri system. Furthermore, it has been widely used in fields such as text categorization [WD08], entity disambiguation [HYB+11] etc. Therefore, apart from its direct visitors, its content is used and accessed implicitly through other sources that are built upon Wikipedia.

The core role and popularity of Wikipedia on the Web and the large variety of applications can be traced to two main factors. First, articles in Wikipedia are constantly evolving and new articles are added by its community of editors. This is mostly influenced by emerging information from the Web. For example, for an existing Wikipedia article like `United States presidential election, 2016`[2], there are news reports, blogs, and other sources reporting about this particular event. In many cases such emerging information is directly reflected in the corresponding Wikipedia articles, thus, keeping Wikipedia up to date. In some cases, real-world events are immediately reflected in Wikipedia within few minutes [KGC11]. Second, due to the Wikipedia policies [pol], especially the *verifiability*[3] policy, it recommends Wikipedia contributors to support their additions with references from authoritative external sources. In particular, this policy states that *"articles should be based on reliable, third-party, published with a reputation for fact-checking and accuracy."*[4]. This policy, on the one hand, guides contributors towards both neutrality and the importance of authoritative assessment and, on the other hand, allows Wikipedia core editors to identify unreliable articles more easily via a lack of such citations. Citations therefore play a crucial role in ensuring and upholding Wikipedia reliability, leading to high quality and important information which is harnessed by the Web users in general and the above mentioned applications.

Despite the established policies and the speed with which editors provide content for Wikipedia articles, these articles vary heavily in terms of quality. First, articles vary in their popularity, hence, their affinity to attract editors and thus provide content. For instance, 51% of Wikipedia articles are categorized as *Stub* or articles in

[1] In 2015 it was in the top 10 most visited Internet sites according to the Alexa Internet ranking www.alexa.com).

[2] https://en.wikipedia.org/wiki/United_States_presidential_election,_2016

[3] https://en.wikipedia.org/wiki/Wikipedia:Verifiability

[4] https://en.wikipedia.rg/wiki/Wikipedia:Identifying_reliable_sources

a need for expansion [Wik]. Second, given that new information regarding articles in Wikipedia constantly emerges, it is hard to keep all articles up to date. Apart from such information overload, the number of active editors[5] at a given time point varies, and is far smaller than the total of 30 million registered editors [Wik]. Naturally, the amount of active editors and other editors demographics and editor's interests will impact the *coverage* of articles that are kept up to date. Furthermore, there is an inherent delay between the time a real-world event happens which is relevant to a Wikipedia article and the time it is reflected in Wikipedia [FAA15]. In addition, changes in a specific Wikipedia article may cause information on other articles to be inconsistent. Finally, for any provided citation in Wikipedia and the text it is cited in, it is not possible to determine the span of text it covers. This has implications in enforcing the *verifiability* policy, where situation may arise in which a paragraph in a Wikipedia article containing a citation may be only partly covered by a reference.

1.2 Scope of the Thesis

Motivated from the importance and wide use of Wikipedia as a resource for a wide range of tasks and its high popularity amongst Web users, we address three core issues which deal with consistency, keeping up to date, and providing trustworthy information for Wikipedia: (i) finding news citations for Wikipedia statements, (ii) citation span determination, and (iii) enrichment of Wikipedia entity pages with novel and important news articles.

Before delving into the details of the problems we address, we clarify some notions we will use throughout this thesis. We will use *Wikipedia article*, with which we refer to Wikipedia *entity* and *event* pages, whereas with *Wikipedia entity page* we refer to only entities. Whereas, with *Wikipedia statement* we will refer to the *piece of text*, ranging from a sentence up to a paragraph, that *has* or *needs* a citation.

(I) Despite the growing trends in terms of the number of entity pages in Wikipedia, and those that adhere to the editing policies in Wikipedia, there is a large set of entity pages whose already existing citations are either outdated or not accessible. Furthermore, as new information is constantly added by Wikipedia editors, it is of great importance to automate or at the very least aid the editors in finding the appropriate citations. Additionally, there are cases in which statements are explicitly marked with *citation needed*; the trust and truthfulness of such statement is into question. This indeed may be the case where the statement is simply not true, however, more often the citation is simply missing and it can be found from sources like *news collections*.

[5] The definition of an *active editor* based on Wikipedia refers to registered users that have contributed in Wikipedia in the last 30 days (for any time point of measurement).

For this problem, where we are given a *piece of text* (we will define later the granularity of the textual fragments that we consider) from a Wikipedia article, we lay out two fundamental research questions.

RQ1.1. *For a* statement *from a Wikipedia entity page, how can we determine the required type of citation (e.g.* news*,* web*,* book *etc.)?*

The outcome from **RQ1.1** is of great importance in finding appropriate citations for any given statement which adhere to the Wikipedia policies. There are many scenarios where specific citation categories are preferred. For instance, for entity pages in the medical domain, a more authoritative reference would be a citation coming from a medical *journal*. In other cases, the availability of specific sources may restrict the statements for which we can suggest a citation.

Next, after knowing the desired citation category of a statement, the problem is how to find such references to cite. This brings us to the second research question which we postulate as following.

RQ1.2. *For a Wikipedia statement that requires a news citation, how can we find news citations which provide evidence for the statement under consideration?*

Automating the process of providing citations as postulated in **RQ1.2** has several advantages. First, it addresses the problem of *long-tail* entity pages, which suffer due to the lack of interest by Wikipedia editors. Second, because Wikipedia is at a constantly evolving state, providing citations in an automated manner will ease the process of editing and serve as a complementary mechanism for Wikipedia editors. Finally, through automation it is possible to enforce in an objective manner the Wikipedia policies without falling into issues that in many cases lead to edit wars and disputability in Wikipedia.

(II) It is evident from the problem in **(I)** that in Wikipedia, determining the granularity for which a citation is valid is somewhat ill-defined. The reasons for this is that there are no explicit requirements and furthermore no means on specifying for what part of text a citation is valid. There are several consequences as a result of this. For instance, if we take a paragraph from a Wikipedia article and a reference cited from a paragraph, we are not able to tell for which part of the paragraph the citation provides evidence for. We formalize the research question addressing this issue as following:

RQ2. *For a paragraph which we extract from a Wikipedia article and a reference cited within the paragraph, how can we accurately determine the span of the citation?*

Determining the span of a citation, that is, singling out at a fine-grained level what a citation covers in a paragraph extracted from a Wikipedia article has important implications. By accurately knowing the span it is possible to have a closed cycle where for uncovered parts in a paragraph we find citation as postulated in **(I)**. Otherwise, for statements which do not have a citation from the appropriate source type, it may be an important signal on the truthfulness and validity of a statement.

(III) Finally, apart from recommending citations for already existing content in Wikipedia article, and finding their corresponding span, a highly important issue remains with emerging information from Web sources regarding a specific Wikipedia article or missing information. Specifically, for a given news collection, we deal with the problem of suggesting news articles to Wikipedia articles, which in turn can be processed by Wikipedia editors in order to add the encoded information within these sources. To this end we postulate the following two research questions.

RQ3.1. *For a Wikipedia entity and a news collection, how can we find news articles in which the entity is a* salient *concept and at the same time the news article provides important and novel information for the entity?*

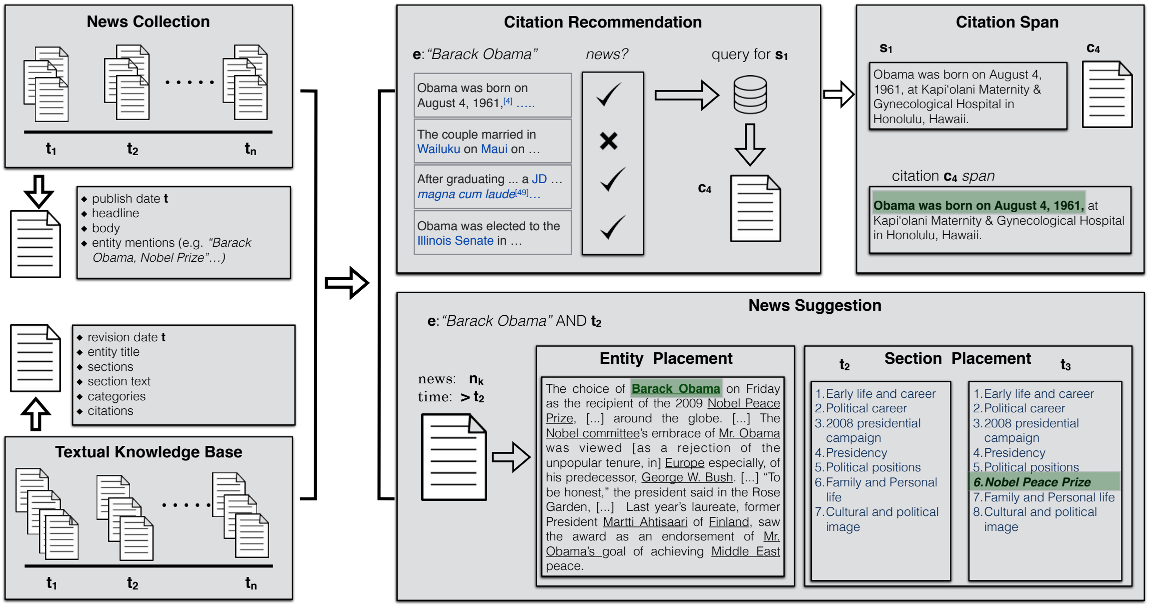
After addressing the question **RQ3.1**, which we refer to as the *article-entity* placement task, we proceed and answer the second question which deals with finding the appropriate *section* in the Wikipedia article.

RQ3.2. *For a Wikipedia entity and a suggested news article, how can we find the* appropriate *section within the entity, and in case such a section is* missing *how can we automatically add the appropriate section in the Wikipedia entity page and* suggest *the news article for?*

The second research question addresses an important issue on suggesting novel information to Wikipedia articles and to specific sections. Due to the fact that information regarding Wikipedia articles constantly evolves, such sections in many cases might be missing. Therefore, to address fully **RQ3.2** one needs to be able to add missing sections for which the news is relevant. For example, for a Wikipedia article `Barack Obama` before his *US presidential election*, the corresponding article did not contain a section about *US Presidency*, hence, in this case, a new section should be suggested automatically, in order to suggest news articles at the appropriate section and at a fine-grained level.

1.3 Contributions of the Thesis

In this thesis, we answer the research questions formalized in the previous section. The contribution of this thesis is on enriching and improving the quality of Wikipedia as one of the most well known textual knowledge bases in the Web. Figure 1.1 shows an outline of our contributions and the proposed solutions for the three core problems listed in Section 1.2.



Obama was born on August 4, 1961,[4] …..

The couple married in Wailuku on Maui on …

After graduating ... a JD … *magna cum laude*[49]…

Obama was elected to the Illinois Senate in …

Obama was born on August 4, 1961, at Kapi'olani Maternity & Gynecological Hospital in Honolulu, Hawaii.

Obama was born on August 4, 1961, at Kapi'olani Maternity & Gynecological Hospital in Honolulu, Hawaii.

The choice of **Barack Obama** on Friday as the recipient of the 2009 Nobel Peace Prize, [...] around the globe. [...] The Nobel committee's embrace of Mr. Obama was viewed [as a rejection of the unpopular tenure, in] Europe especially, of his predecessor, George W. Bush. [...] "To be honest," the president said in the Rose Garden, [...] Last year's laureate, former President Martti Ahtisaari of Finland, saw the award as an endorsement of Mr. Obama's goal of achieving Middle East peace.

Figure 1.1. Overview of the proposed approach for enrichment and improvement of textual knowledge bases. The approach shows the three main steps: (i) *Citation Recommendation*, (ii) *Citation Span*, and (iii) *News Suggestions*.

(I) ***News Citation Recommendation for Wikipedia*:** In Chapter 5 we propose a novel approach for finding *news citations* in Wikipedia. We address the two research questions in problem **(I)**.

- **RQ1.1.** Firstly, for a Wikipedia article and a specific *statement*, we propose an approach which determines the *type* of resources that are appropriate for citation. Dependent on the statement at hand, there is a range of 12 citation types (e.g. *news, web, journal, book, report, . . .*) which can be chosen from. This is a prerequisite for finding appropriate citations that enforce the Wikipedia editing policies, where *authoritative* sources are suggested. Which source type is considered more authoritative is dependent on the article and the statement, however, if an article is about medicine a source from a journal is preferred over a news article. To determine the citation category, we rely on *language style* and other *structural* attributes we extract from Wikipedia articles. This addresses question **RQ1.1**.

- **RQ1.2.** Second, we focus on only the cases where for any piece of text in Wikipedia the required and appropriate citation is of type news. The reason for focusing on this specific type of citation is motivated in Chapter 3. We automatically construct a query to find news articles from a news corpus. We determine which news article to suggest as evidence for the given statement based on *textual entailment*, *news authority* and other *centrality* measures.

The contributions from this chapter are published in:

- [FMNA16] Besnik Fetahu, Katja Markert, Wolfgang Nejdl, Avishek Anand: Finding News Citations for Wikipedia. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016, pages 337–346.

(II) ***Fine-grained Citation Span for References in Wikipedia*:** In Chapter 6 we propose an approach, where for a paragraph containing a *web* or *news* citation we determine the coverage span of the citation. The work is of importance for automated approaches on enriching and expanding Wikipedia. In this way, we can determine to what extent an entity page adheres to the *verifiability* policy. Furthermore, we provide explicit markings of the statements in Wikipedia that are covered by a citation. Hence, tying this together with the contribution in Chapter 5 we close the cycle in which we find citations for uncovered statements until for all statements we can provide citations, in case they exist in a given news collection.

The contribution in this chapter has been published in:

- Besnik Fetahu, Katja Markert, Avishek Anand: Fine Grained Citation Span for References in Wikipedia. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP 2017, Copenhagen, Denmark, September 7-11, 2017 (to appear).

(III) ***Automated News Suggestion for Populating Wikipedia Pages*:** In Chapter 7 we propose a novel approach for accounting for the ever evolving nature of Wikipedia entity pages, respectively, the emerging information in news and Web sources in general. Furthermore, due to the varying popularity of Wikipedia articles and respectively their affinity to attract editors to provide content for such articles, we can retrospectively suggest missing information for such articles. The proposed approach addresses the following research questions:

- **RQ3.1.** First, for a news article, we consider the dual problem of determining the *salient* entities in the news article and consequentially if the news article is of importance for the Wikipedia entity page at hand. In this way, we ensure that only important information is suggested where the entity is a salient concept. This addresses the research question in **RQ3.1**.

- **RQ3.2.** Second, considering the section structure of Wikipedia articles, it is important to determine precisely for which section a suggested news article is relevant. Due to the fact that Wikipedia articles evolve with new information becoming available, and along with that the section structure will change. Therefore, in our proposed approach we account for the two attributes, by first determining the appropriate section for which we suggest the news article, and in case such a section is missing we suggest its addition into the section structure.

The contribution in this chapter has been published in:

- [FMA15] Besnik Fetahu, Katja Markert, Avishek Anand: Automated News Suggestions for Populating Wikipedia Entity Pages. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM 2015, Melbourne, Australia, October 19 - 23, 2015, pages 323–332.

Apart from our holistic approach on dealing with enrichment and improvement of Wikipedia articles, we additionally make the following contributions which provide the context for the work carried in this thesis.

(A1) ***How much is Wikipedia lagging behind News?*** The importance of news in Wikipedia is acknowledged by its editing policies [pol], where authoritative and third-party sources like news articles are suggested for citation. To better understand the actual use of news as citations in Wikipedia we analyze how news are reflected in Wikipedia, respectively, the amount of time it takes for an entity to be reported in news and its occurrence in Wikipedia, and finally, how many of the citations from all citation categories are news citations.

The importance and interaction between news and Wikipedia is published in:

- [FAA15] Besnik Fetahu, Abhijit Anand, and Avishek Anand. How much is wikipedia lagging behind news. In *Proceedings of the ACM Web Science Conference, WebSci 2015*, Oxford, United Kingdom, June 28 - July 1, 2015, pages 28:1–28:9.

(A2) ***Improving Entity Retrieval in Structured Data.*** Finally, we look into application use cases, where Wikipedia drives several major industry projects on constructing knowledge graphs by Google [DGH+14], Yahoo! [BMV11], Microsoft [NMS+07] which drive the functionalities behind entity search.

In this case, we address several of the shortcomings that mostly deal with the nature of such structured datasets (e.g. DBpedia, Freebase etc.) and propose a new query similarity model for entity search. The contribution of this work has been published in:

– [FGD15] Besnik Fetahu, Ujwal Gadiraju, and Stefan Dietze. Improving Entity Retrieval on Structured Data. In *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference*, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I, pages 474–491.

2 Foundations and Technical Background

In this chapter, we introduce the technical background necessary to understand the work carried in this thesis. We first introduce the notion of knowledge bases, then continue on entity linking techniques. Next, we provide a thorough analysis of information retrieval techniques. We then describe clustering techniques, and finally conclude with supervised learning and feature selection algorithms.

2.1 Knowledge Bases

With the term *knowledge base* (KB) we refer to RDF datasets published based on a set of *linked data* principles introduced by Berners-Lee et al. [BLHL$^+$01]. Since then there has been a big push towards publishing data according to these principles. Nowadays, the number of datasets is in the range of thousands [LS]. KBs are commonly referred to as *structured datasets, linked datasets* or *RDF datasets.*

2.1.1 Resource Description Framework – RDF

The term knowledge bases (KB) refers to RDF datasets published based on linked data principles introduced by Berners-Lee et al. [BLHL$^+$01]. This can be considered also as the inception of the field of Semantic Web. Since then, there has been a big push towards publishing data following the principles, known as the *linked data* principles, introduced in [BLHL$^+$01]. Nowadays, the number of datasets is in the range of thousands [LS]. These datasets are represented in RDF format and are interchangeably referred to as *structured datasets, linked datasets* or *RDF datasets.*

Resource Description Framework, or RDF, is a graph data model proposed by W3C as a standard for knowledge representation [W3Ca]. The RDF data model consists of a set of *resources*, *predicates*, and *literals.*

A *resource* refers to a real-world entity (e.g. person, organization etc.) or an abstract concept. We denote with $\mathcal{R}$ the set of resources in a KB. *Literals*, $\mathcal{L}$, on the other hand, represent values like a string, date, number etc. Finally, *predicates*, $\mathcal{P}$, represent a relation between resources or a resource and a literal.

We define a KB to be the projection between these building blocks of the RDF data model. Hence, a knowledge base can be represented as following: $\mathcal{K} := \mathcal{R} \times \mathcal{P} \times (\mathcal{R} \cup \mathcal{L})$. Alternatively, a KB can be seen as simply a set of triples of the form $\langle s, p, o \rangle$, where $s \in \mathcal{R}$, $p \in \mathcal{P}$, and the object o can represent a resource or a literal, thus, $o \in \mathcal{L} \cup \mathcal{R}$.

Listing 2.1 shows a set of triples describing a resource, which here represents $s :=$ `db:University_of_Hannover`.

Listing 2.1 RDF Resource Example for *"University of Hannover"*

```
db:University_of_Hannover rdf:type    owl:Thing .
db:University_of_Hannover rdf:type    dbo:Agent .
db:University_of_Hannover rdf:type    dbo:EducationalInstitution .
db:University_of_Hannover rdf:type    dbo:Organisation .
db:University_of_Hannover rdf:type    dbo:University .
db:University_of_Hannover dbo:city    db:Hannover .
db:University_of_Hannover dbo:country db:Germany .
db:University_of_Hannover dbp:budget  "441.8 million"@en .
```

2.1.2 RDF Schema – RDFS

A crucial construct in publishing RDF datasets is the organization of resources into *classes*. This functionality is provided by RDFS [W3Cb]. RDFS is an extension of the basic constructs provided by the RDF data model. A class in RDFS is defined by the set of triples in Listing 2.2. A resource s is assigned to a class c by the triple $\langle$ s, `rdf:type`, $c\rangle$, similar to Listing 2.1 where $\langle$ `db:University_of_Hannover`, `rdf:type`, `owl:Thing`$\rangle$.

Listing 2.2 RDFS Class Definition Example

```
dbo:University    rdf:type           rdfs:Class .
dbo:University    rdfs:subClassOf    dbo:Organisation .
dbo:Organisation  rdfs:subClassOf    dbo:Agent .
dbo:Agent         rdfs:subClassOf    owl:Thing .
```

RDFS allows us to express hierarchy relations between different classes through `rdfs:subClassOf`. As such, the resource `db:University_of_Hannover` assigned to class `dbo:University`, by traversing the class hierarchy we assume that it is also of type `owl:Thing`.

2.1.3 Real-World Knowledge Bases

The functionalities of RDF and RDFS, and similar Semantic Web initiatives on modeling and representing knowledge have resulted in many initiatives on representing data according to such standard and principles. Arguably, some of the most well known examples, include knowledge bases like DBpedia [BLK+09], YAGO [SKW07]. Such KBs represent a subset of the information contained in Wikipedia articles, and represented according to linked data principles [BLHL+01].

These KBs are particularly interesting for this thesis. They allow to construct homogeneous groups of Wikipedia articles based on a type taxonomy constructed for Wikipedia articles based on the category structure in Wikipedia. Such categorization of articles according to a type taxonomy is useful for the approach in Figure 5.1. In many cases articles that are of different types, e.g., articles about *Places* and *Politicians*, have completely different structure and the problems we tackle behave differently for the different types. Therefore, treating such articles separately accounts for accurate and reliable models.

2.2 Entity Linking and Disambiguation

In this section, we present an overview of state-of-the-art approaches on entity linking (EL) and disambiguation (NED) techniques. The task here corresponds to canonicalizing *surface forms* or *text phrases* to entities on a given database of entities. In majority of the cases [HYB+11, MJGB11, FS12] Wikipedia is used as the target to link such surface forms to entities.

The core problem in this task is to resolve *ambiguous* mentions of entities in free text. Figure 2.1 highlights the problem of resolving entity mentions from a text snippet into entities in Wikipedia.

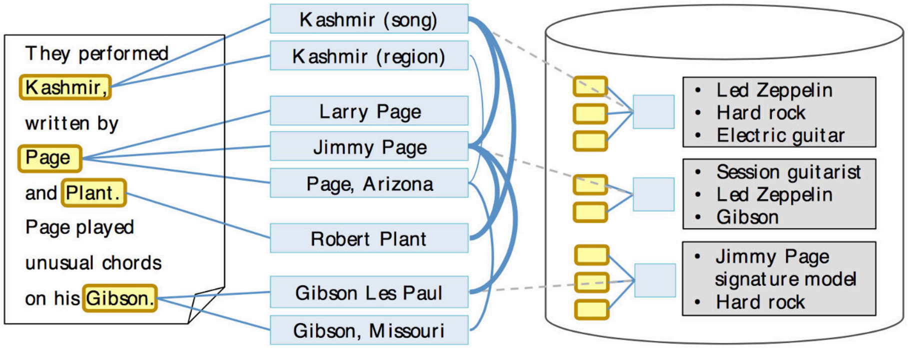


Figure 2.1. Mention-Entity graph for an example text snippet with ambiguous entity mentions [HYB+11].

We highlight two main differences between state of the art approaches on NED and EL. In NED systems, the candidate mentions from a text snippet are limited to only those that resolve to a named entity of type {*Person, Location, Organization, Time*}. This is usually done through named entity recognition (NER) approaches like [FGM05]. Contrary to NED approaches, in EL there is no restriction in the mentions that can be resolved to entities on a target knowledge base. For instance, through EL systems one can resolve a mention referring to an event, e.g. *U.S Presidential Elections 2016* to its corresponding Wikipedia article[1], which through NED approaches is not possible.

In this thesis, we opt for entity linking approaches due to their wider coverage of linking mentions from news and Web sources to Wikipedia articles. While the different entity linking and disambiguation approaches differ on their final result of disambiguated or linked entities, however, on common attribute of both applications is that they consider the *coherence* and *contextual* similarity between entity candidates in a textual snippet as part of their linking or disambiguation results.

NED. AIDA [HYB+11] is a state of the art approach on named entity disambiguation. For a text snippet as shown in Figure 2.1, AIDA performs the following operations to accurately link ambiguous entity mentions into a KB. Based on a pre-processing step where through NER are extracted mentions to named entities it generates a list of entity candidates from the target KB. Finally, the disambiguation is performed jointly by constructing a graph of mentions and entity candidates from the target KB. The goal is to find a dense sub-graph that fulfills the following properties: (i) high *contextual similarity* between mentions in the text and the candidate entities in KB, (ii) weighted edges amongst the candidate entities, which measure their *coherence*. The coherence in this case is a function of the number of incoming links two entities share in a KB. Common incoming links for any two entities e_1 and e_2 in a KB can be defined by the following triples $\langle x, p, e_1\rangle$ and $\langle x, p, e_2\rangle$.

EL. The work by Milne and Witten [MW08] is one of the most notable works and serves as the basis for many EL approaches. It relies on *anchor text* from Wikipedia articles which are used to learn models, and are later applied on textual resources to link specific phrases to Wikipedia articles. A commonality between AIDA and this work is the coherence between entities [WM08] and is computed occurring in a textual snippet, and contextual similarity. The coherence score is computed as in Equation 2.1.

TagMe [FS12] is a state of the art in entity linking. The advantage of TagMe compared to other existing approaches, is that it optimizes for short textual snippets as well. This provides an advantage considering that a significant proportion of news and web resources in general are not very lengthy. TagMe follows a similar scheme on performing the entity linking, however, with improvements on disambiguating surface

[1] https://en.wikipedia.org/wiki/United_States_presidential_election,_2016

forms that map to anchor texts in Wikipedia and their corresponding articles based on a voting scheme. In the voting scheme, for each anchor text and a candidate article it is computed an average relatedness score (see Equation 2.1) w.r.t the other articles that are candidates from other existing anchors in a textual snippet. Finally, to account for the efficiency and accuracy of the approach, the candidate articles are pruned based on the link probability (the score that an anchor text links to a Wikipedia article) and the voting score for each anchor and Wikipedia article pair.

$$rel(e_1, e_2) = \frac{\log\left(max(|E_1|, |E_2|) - \log(|E_1 \cap E_2)\right)}{\log(|E|) - \log(min(|E_1|, |E_2|)} \tag{2.1}$$

where E_1 and E_2 represent the set of incoming links to entities e_1 and e_2, respectively. E represent the set of all entities in a KB.

2.3 Information Retrieval

Information Retrieval (IR) deals with the means on accessing and satisfying user information needs through querying of large collections, mostly of unstructured documents. Despite its foundations being on unstructured documents, IR has become a multi-modal field, providing techniques for access of multimedia objects and other structured datasets like KBs.

In this thesis, we will discuss relevant *query models* within the scope of this thesis, respectively for textual collections and structured datasets. The main idea behind any query similarity model is the following.

For a document collection $\mathcal{D}$ which is projected into a vocabulary space of terms $\mathcal{V}$, and a query $q \in \mathcal{V}$, the task is to find relevant documents from $\mathcal{D}$ such that they satisfy the information need in q.

From the task above we highlight two key points: (i) representation of documents and queries, and (ii) document relevance for a given query.

2.3.1 Document and Query Representation

The de-facto representation of documents and queries is based on the proposed vector space model by Salton et al. [SWY75]. A document is represented by the terms occurring in the document, and for each term we can assign boolean indicator values or some form of weight reflecting the importance in the document. Similarly, queries are represented into the vector space model. Through this representation it is easy to compute the relevance of a document for a given query. This leads to ranked retrieval, where the documents are ranked according to their relevance to the query. Figure 2.2 shows a three-dimensional [SWY75] representation of a document collection and the similarity between documents based on the dot product of the vector representations.

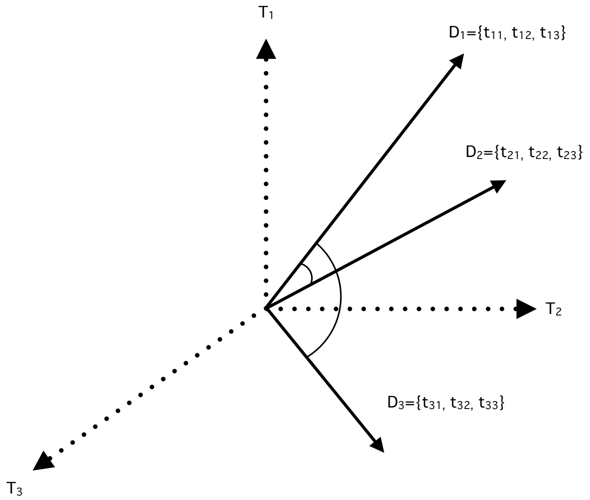


Figure 2.2. Vector representation of a document collection. Each term is drawn from a vocabulary and it represent a dimension in the document representation.

Here, it is assumed that the vector space representation is drawn from a vocabulary of terms $\mathcal{V}$ which corresponds to all the terms (words, or stemmed words) in the given document collection $\mathcal{D}$.

Hence, a crucial part in computing the similarity between a query and a document based on their vector space representations is the assignment of the weights for the term occurrences. The most widely used weighting scheme is based on the *tf-idf* [MRS08]. That is, the *term frequency* or tf measures the frequency of a term $v \in \mathcal{V}$ in a document $d \in \mathcal{D}$, whereas the *idf* or *inverse document frequency* counts the number of documents in which the term v occurs. Through tf it is measured the importance of term for a document, whereas with *idf* it is measured how well such a term distinguishes a document from others. Their combination yields the trade-off between the two, and its simplest variation is computed as in Equation 2.2.

$$tfidf = tf(v,d) \cdot \underbrace{\frac{|\mathcal{D}|}{df(v)}}_{\text{idf}} \tag{2.2}$$

where, $df(v) = |d \in \mathcal{D} : v \in d|$, representing the number of document in $\mathcal{D}$ containing term v.

Through the $tfidf$ term weighting, document retrieval becomes a function of ranking documents based on the sum of the query terms present in a document d. For example, a basic ranked retrieval of documents is shown in Equation 2.3.

$$r(q,d) = \sum_{v \in q} tfidf(v,d) \tag{2.3}$$

2.3.2 Information Retrieval Models

Okapi BM25. One of the most widely used retrieval models is BM25 [RWJ+95]. Contrary to the model based solely on the *tfidf* scores, BM25 requires parameter tuning that are dependent on the given document collection $\mathcal{D}$. Furthermore, in its query-document scoring function it takes into account the document length. The BM25 scoring model is shown in Equation 2.4.

$$r(q,d) = \sum_{v \in q} w_{tf}(q,d) \cdot w_{idf}(v) \tag{2.4}$$

where, the term frequency score $w_{tf}(q,d)$ is computed as following:

$$w_{tf}(q,d) = \frac{(k_1+1)\cdot tf(q,d)}{k_1 \cdot \left((1-b) + b \cdot \frac{len(d)}{avglen}\right) + tf(q,d)} \tag{2.5}$$

where, the parameters k_1 ($k_1 \geq 1$) and b ($0 \leq b \leq 1$) are tunable, and are usually set to values $k_1 = 1.2$ and $b = 0.75$, respectively. Here, b controls how much we normalize the term frequency scores according to the document length and its ratio to the average document length in $\mathcal{D}$. With $len(d)$ and $avglen$ we note the length of document d, and average document length in $\mathcal{D}$, respectively.

The inverse document frequency score w_{idf} for a query term is computed as following:

$$w_{idf}(v) = \log \frac{N - df(v) + 0.5}{df(v) + 0.5} \tag{2.6}$$

here, N represents the number of documents in $\mathcal{D}$, and $df(v)$ as defined above represents the number of documents containing term v.

Divergence from Randomness – *DFR*. Contrary to BM25, DFR is a non-parametric approach [AVR]. The advantages of non-parametric models is that there is no need to rely on expensive parameter tuning approaches or have collection specific parameters.

In *DFR*, contrary to probabilistic models like BM25, where the documents are ranked based on their relevance, here the documents are ranked according to the *gain* in retrieving a document given a query term. The basic idea behind *DFR* is the weighting of the query terms for a given document collection. The weighting is done according to the scheme in Equation 2.7 which computes two probability distributions.

$$w(v) = -\log_2 {P_1}^{(1-P_2)} \tag{2.7}$$

The first probability distribution P_1 is a function of the within-document term frequency and reflects the chance of having a specific term frequency of a term v in a document d. The intuition here is to measure the information gain of a specific term w.r.t a document collection. The higher the probability P_1 for a term the lower its gain. This is usually refers to the notions of *"nonspeciality"* and *"speciality"* words on a given document collection [Har75]. Hence, for P_1 we have $-\log_2 P_1$ which is a monotonically decreasing function and represents how informative a term is for a document w.r.t a collection.

In the second probability distribution P_2 it is measured the likelihood of the term occurrence only to a subset of documents in a collection called the *elite documents*. This set can differ on how it is constructed. For instance, in Harter et al. [Har75] this is defined as the set of documents which are important for a given term, while in [AVR] this is simply represented by the documents that contain a given term. Finally, the intuition here captures how *informative* a term is, that is, the lower the expected score of P_2 the higher the gain of the term is as it can be seen from Equation 2.7.

Finally, the two probability distributions are subject to the term frequency, the size of the document collection, the size of the elite set, and the frequency of a term in the elite set. Whereas, the probability models or the randomness models usually range from the *Binomial distribution*, *Poisson distribution*, etc. [AVR].

2.4 Clustering Approaches

In this section, we describe clustering approaches which are used in this thesis. Clustering techniques are often used in the context of unsupervised learning, where data items are grouped together based on a pre-defined notion of similarity. In some cases, clustering is used to improve efficiency of algorithms, where pair-wise similarity is required and as such by limiting the comparisons only to a limited set of data items provides drastic gains in terms of efficiency.

Here we will discuss three main clustering techniques. First, we describe the most fundamental clustering algorithm *k–means* [HW79] and a more sophisticated version of it, namely *X–means* [PM$^+$00]. Next, we discuss *Locality Sensitive Hashing* (LSH) [LRU14] which clusters data items based on generated min-hash signatures. Finally, we describe *spectral clustering* [vL07] a clustering algorithm which relies on matrix factorization approaches.

2.4.1 k–means Clustering

k–means or Lloyd's algorithm [HW79] is one of the first and most widely used clustering algorithms. In this algorithm, data items with a multidimensional representation, where this representation can be arbitrary (e.g. document terms, feature values), are grouped into k clusters. In the first step, the algorithm chooses k random *cluster centroids*, which correspond to random data items, and data items are associated with one of the clusters based on the minimal Euclidean distance between the centroid and the data item. The process is iterative, where after each iteration the cluster centroids are updated with the new representation from all its data items assigned to it, and the data items are re-assigned to their closest centroids. The process stops when the data items do not change their cluster assignment.

The quality of clustering is measured by the *residual sum of squares* (RSS) which captures the distance from a data item to its cluster centroid.

$$RSS_k = \sum_{\vec{x} \in \omega_k} |\vec{x} - \vec{\mu}(\omega_k)|^2 \tag{2.8}$$

where $\vec{x}$ represents an item assigned to cluster ω_k, whereas $\vec{\mu}(\omega_k)$ represents the cluster centroid.

The main disadvantage of the *k–means* algorithm is the choice of the k for the number of clusters, and the initial assignment of the cluster centroids. For the latter issue, there are approaches that consider the inter-centroid distance in the initial step, trying multiple initial cluster assignments, or initial centroids which are based on prior information [BF98].

2.4.2 X–means Clustering

Pelleg and Moore [PM+00] proposed *X–means*, a clustering technique which overcomes three main drawbacks of the standard *k–means* algorithm. First, it provides a more efficient algorithm for performing the clustering process. Second, it automatically finds the right number of clusters, which optimize the *Bayesian Information Criterion* (BIC). Finally, it ensures that the clustering results do not fall into *local minima*, as is the case for *k–means*.

To find the right k for clustering, as input it is required the *lower* and *upper* bound of the search space for which the k can take values. The algorithm consists of three steps: (i) run the conventional *k–means* for a specific k, (ii) improve the structure of the clusters by estimating whether new clusters should appear within the already existing k clusters, that is, increase the number of k, and (iii) if $k > k_{max}$ report the best scoring model for BIC.

X–means for a given k_{max}, in the worst case tries all $2^{k_{max}}$ configurations. That is, for each cluster, it considers if the BIC score is improved by further splitting into multiple clusters. The rate of splitting is determined by how close the current setting is to the true distribution of the data.

In *X–means*, BIC is computed as in Equation 2.9. M_j corresponds to a clustering solution for a specific k, D represents the data points, $|D|$ is the number of data points, p_j is the number of parameters in the model M_j, finally $\hat{l}_j(D)$ represents the log-likelihood of the data according to the model M_j.

The estimation of the log-likelihood of the data is done under the assumption that the models are *spherical Gaussians*, which is the assumption made by the *k–means* algorithm. The log-likelihood $l(D)$ is shown in Equation 2.10, whereas by taking the derivative $\hat{l}$ we obtain the *maximum log-likelihood*.

$$BIC(M_j) = \hat{l}_j(D) - \frac{p_j}{2} \cdot \log |D| \tag{2.9}$$

$$l(D) = \log \prod_i P(x_i) = \sum_i \left(\log \frac{1}{\sqrt{2\pi}\sigma^M} - \frac{1}{2\sigma^2} ||x_i - \mu_{(i)}||^2 + \log \frac{|D_{(i)}|}{|D|} \right) \tag{2.10}$$

the estimation of the variance σ for a model M_j with the assumption that the data is distributed under the spherical Gaussian model.

2.4.3 Minhashing and Locality Sensitive Hashing

When dealing with a large set of documents or data items, for different clustering approaches a fundamental issue remains efficiency. For instance, sophisticated clustering approaches like *spectral clustering*, computing *adjacency matrices* or the pairwise similarity between documents is highly expensive. This directly correlates to the number of documents, e.g. for 1000 documents the number of pairwise comparisons is already $\binom{1000}{2}$, with nearly 500k comparisons.

To improve on efficiency, there are approaches that construct signatures of documents and project them into a space where documents that may have slight similarity have similar representation. This gives rise to the computation of *minhash signatures*, and respectively *locality-sensitive hashing* clustering which uses the generated *mihash* signatures in order to group likely similar items together.

Below we describe two techniques which allow to improve on efficiency of clustering large collections [LRU14].

Minhash Signatures. To avoid computation of pair-wise similarities between documents in a large collection, we need to be able to have compact representations of such documents to leverage it for clustering. *Minhash signatures* compute the *characteristics matrix*, which as rows contains the document representation (i.e. terms) and as columns the documents in a collection. A toy example from [LRU14] is shown in Table 2.1. Next, the minhash signature of a document from the characteristic matrix is the number of the row from the permuted set of terms representing the document collection where we encounter a term contained in a given document. For example the minhash signature of $h(d_1) = v_3$.

A highly useful property of the compute minhash signatures is that for a random permutation of the terms representing the document collection, the probability that the minhash function produces the same signature for two documents is equal to the *Jaccard* similarity.

item	d_1	d_2	d_3	d_4
v_1	0	0	1	0
v_2	0	0	1	0
v_3	1	0	0	1
v_4	1	0	1	1
v_5	0	1	0	1

Table 2.1. A characteristic matrix representing a set of documents (in the columns) and for the terms drawn from the vocabulary of terms from all documents.

	d_1	d_2	d_3	d_4
h_1	1	3	0	1
h_2	0	2	0	0

Table 2.2. Minhash signatures after mapping the rows into the hashing buckets based on h_1 and h_2 and after replacing each hash column with the lowest hash bucket for each document for all its non-zero entries.

Enumerating all possible permutations of the representation space is expensive. One way to remedy such a problem and pick a random permutation of the rows representing a document collection is to map the rows through a set of hash functions. The hash functions map the rows into as many buckets as there are rows. Here one precaution to take is that the number of collisions or mapping of different rows into the same bucket is not high.

Instead of computing random permutations, we map the rows through a set of hash functions into random buckets (where the number of buckets is ideally as many as the number of rows). The minhash signatures are computed as following: for each document the signature corresponds to the number of hash functions, and for each column where the document has a specific term, in the signature each value is replaced with the lowest hash value mapping the rows to the random buckets.

Considering the same example as in Table 2.1, and if we take two hash functions $h_1(x) = x + 1 \texttt{ mod } 5$ and $h_2(x) = 3x + 1 \texttt{ mod } 5$, with which we map the rows into their corresponding hash buckets, we have the following minhash signatures in Table 2.2.

Locality Sensitive Hashing. Through *minhash signatures* we can compress the representation of large collection while preserving the similarities between documents. However, we are still left with a combinatorial problem of pair-wise comparisons between documents.

One way to remedy this problem is through the *Locality Sensitive Hashing* (LSH) approach. It groups together documents that are up to a certain degree similar. We can do this by splitting the minhash signatures into b bands or buckets consisting of k rows each. For each band we associate a hash function which takes the k rows from each band and for each item and generates a hash value corresponding to some bucket. Similar vectors will be bucketed together by the same hash function. Furthermore, since the minhash signatures are generated from random permutations of the characteristic matrix, in one of the bands if there is any similarity between any two document they will be bucketed together by one hash function.

For any given threshold of Jaccard similarity s for which two documents are considered candidates for pair-wise comparison, there are probabilistic guarantees that they agree on all rows in a band b is s^r, whereas that they disagree in at least one row is $1 - s^r$. Finally, we can generalize this and provide the probability of the signatures agreeing on all rows in at least one band b with $1 - (1 - s^r)^b$.

2.4.4 Spectral Clustering

Spectral clustering is a popular clustering algorithm due to its ability to group data points of any arbitrary shape. Other algorithms like *k–means* make strong assumptions on the form of the clusters, hence, they are not flexible in detecting arbitrary shapes like *spectral clustering*. Furthermore, through *spectral clustering* we do not get stuck in local minima and there is no need to restart the clustering process with different cluster centroids to obtain optimal results [vL07].

We explain the intuition behind these favorable properties of *spectral clustering*. In spectral clustering, for any given set of data items (documents or any other type of data) we perform the following steps to compute the clusters:

Similarity Graph. Assume we have a collection of documents $\mathcal{D}$. First, we construct the *similarity graph*, $G = (V, E)$ where the vertices correspond to the documents $d \in \mathcal{D}$ and the edges are between documents which have some notion of similarity $s(d_i, d_j)$. In spectral clustering, there are mainly three approaches on generating the graph G: (i) ϵ–neighborhood graph consists of all connected vertices whose distance based on $s(d_i, d_j)$ is less than ϵ, (ii) k–nearest neighbors, with the graph consisting of the k nearest vertices for d_i, and (iii) *fully connected graph* where all the vertices are connected if their similarity is above zero, and the edge weights correspond to the similarity score s.

In the case of the similarity graph it is important the choice of the similarity measure in order to get meaningful clusters, however, one pre-condition for applying spectral clustering is such that the similarity $s(d_i, d_j)$ should be always positive and should be a symmetric measure.

Graph Laplacians. From the constructed similarity graph G, we compute the *graph Laplacians*. There are two main graph Laplacians that are used in spectral clustering: (i) *unnormalized* and (ii) *normalized* graph Laplacians.

The unnormalized graph Laplacian $L = D - W$, where D is the diagonal matrix, where each entry in the diagonal matrix corresponds to the sum of weights or similarities between all the items in that row, $d_i = \sum_j s_{ij}$ and W is the adjacency matrix computed for data under consideration. Whereas, the normalized graph Laplacian is computed as $L_{sym} = I - D^{-1/2}WD^{-1/2}$, where I represents the *identity matrix*.

Clustering. From L (independent of which one is used), compute the first k *eigenvectors* $(u_1, u_2, \ldots, u_k)$, which results in the matrix $U \in \mathbb{R}^{n \times k}$. Now from the computed eigenvectors from all the data points, usually a simple *k–means* clustering algorithm is employed to cluster the items based on their eigenvectors, where as a similarity measure between such vectors is used the Euclidean distance.

Similarly as for many other existing clustering algorithms, an issue here is to determine the right number of clusters. However, there are simple heuristics that leverage the eigenvector space to find the right number of clusters. One common technique is to consider the distribution of *eigenvalues*, such that the number of clusters k corresponds to the first k eigenvalues, that is, $\lambda_1, \ldots, \lambda_k$ are very small, but λ_{k+1} is relatively high.

2.5 Supervised Learning and Feature Selection

In this section we describe supervised learning approaches, and additionally distinguish a class of the so called *structured prediction* in supervised learning. Finally, we describe how we can quantify the importance of features which are used in the learning algorithms.

2.5.1 Supervised Learning

For learning, we assume the setting where we are given a set of data items represented in an n–dimensional feature space. That is $X \in \mathbb{R}^{k \times n}$, where $X = \langle X_1, X_2, \ldots, X_k \rangle$. The task is to predict a discrete output Y, hence, we can generalize the learning approaches as to learning the function $f : X \rightarrow Y$.

Logistic Regression – LR. It is one of the most simplistic and widely used supervised learning algorithms [Bis06]. For a set of training data X it learns n feature weights based on the *maximum likelihood principle* (MLP). The classification model is represented by a linear function which for an instance $X_i \in X$ chooses the $y \in Y$ that maximizes the probability $P(Y = y|X_i)$.

LR estimates the likelihood of an instance X_i belonging to a class $Y = y$ as shown in Equation 2.11. The most probable class is chosen by simply taking the $y \in Y$

which maximizes the probability $P(Y = y|X_i)$.

$$P(Y = y|X_i) = \frac{exp(\theta_y + \sum_{j=1}^{n} \theta_{y,j} X_{i,j})}{\sum_{i=1}^{k} exp(\theta_y + \sum_{j=1}^{n} \theta_{y,j} X_{i,j})} \tag{2.11}$$

where $\theta_{y,j}$ $(j = 1, \ldots, n)$ represents feature weights that are estimated based on MLP for a given set of training data. This is done by providing θ as a free parameter, namely $\theta \leftarrow \text{argmax}_\theta \prod_{i=i}^{k} P(Y_i|X_i, \theta)$.

While logistic regression has found wide adaptation for many classification tasks, one main disadvantage is its linearity, that is, it can classify accurately instances that are only linearly separable.

Random Forests – RF. They belong to a widely technique of supervised learning, namely *ensemble learning* [Bre01]. In Random Forests (RF), the input space from a set of training instances is split into K classification trees resulting into a *forest*. The generation of trees follows two main principles. First, the input feature space is split into K random vectors, or feature subsets chosen randomly, resulting in classification trees h_K. In the second approach, in the case of low-dimensional input feature space, one can employ a *linear combination* of features and consequentially generate classification trees based on the CART algorithm [BFOS84].

The classification in RFs is performed based on the *majority voting* scheme. That is, for a given instance X_i we generate K labels from the K trees in our random forest and finally pick the label y which was predicted by the majority of K trees.

RFs [Bre01] provide theoretical guarantees on the performance of the classifiers, which is measured through a *margin function* shown in Equation 2.12.

$$margin(X, Y) = avg_k \mathbb{I}(h_K(X) = Y) - \max_{Y' \neq Y} \mathbb{I}(h_k(X) = Y') \tag{2.12}$$

The margin measures the distance in number of votes from the correct label for an instance versus the incorrect label which has the highest votes on average.

Support Vector Machines – SVM. Are a widely used supervised learning approach when the input feature space for training instances it is high. Introduced first by Vapnik [CV95], the task is to construct a *hyperplane* which separates input instances linearly. An optimal hyperplane is constructed based on so called *support vectors*, which determines the maximal margin between support vectors of different classes. Figure 2.3 shows an example of support vectors for an optimal hyperplane.

In more details, in SVMs for linear model $y(x) = \mathbf{w}^T\phi(X)+b$, where $\phi(X)$ denotes the multi-dimensional representation of instance X, we find the optimal weights $\mathbf{w}$ and parameter b based solely on the support vectors. In this case, based on the linear model $y(x)$ the support vectors satisfy the equation $y_i(\mathbf{w}^T\phi(X) + b) = 1$. In order to find the optimal weights, the task is to find the support vectors that maximize the distance between support vectors of two classes. Furthermore, in SVMs, through a *kernel trick*, it allows for feature spaces that are larger than the number of instances to be classified efficiently. In cases where the instances are not linearly separable one can employ non-linear kernels which maps the feature space into a non-linear one where the classification can be performed accurately.

The optimal weights are estimated subject to the support vectors and are discussed in details in [CV95, Bis06].

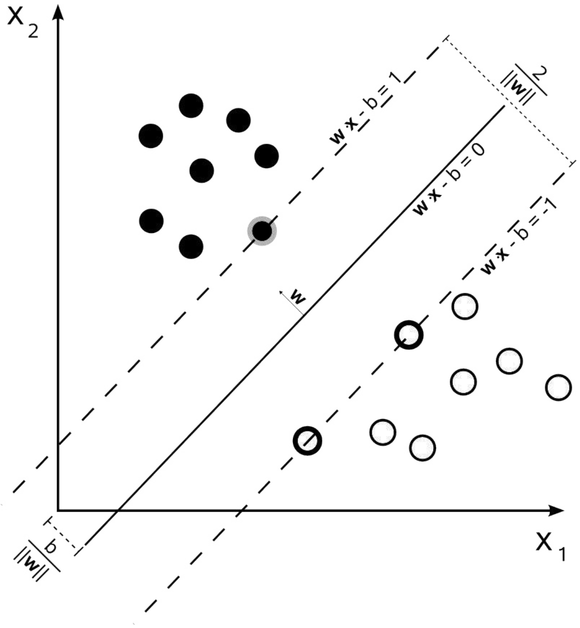


Figure 2.3. An example of linearly separable instances. The support vectors are the ones which are at the margins and they define the maximum distance between the support vectors between the two classes.

Finally, in the case of multi-class classification problems, SVMs consider two schemes which convert the multi-class classification into a binary classification problem. The first scheme refers to as *one-versus-the-rest* (OvA), where the class of interest is considered as the positive class, whereas the others as negative classes. The class with the highest confidence based on the OvA scheme is assigned as the class of an instance. The second scheme, *one-versus-one* (OvO) trains $\binom{K}{2}$ classifiers for all pairs of classes, and classifies an instance based on the majority voting scheme from the resulting classifiers.

2.5.2 Structured Prediction

The main difference between discrete classification models (previously described approaches) and *structured prediction* are the following. In the first case, we optimize for classifying an instance or data point to a discrete output, e.g. binary label or any multinomial variable. In structured prediction, the task is to predict jointly a sequence of instances or data points. That is, the instances in a sequence are dependent on each other and thus the name structured prediction aka sequence classification.

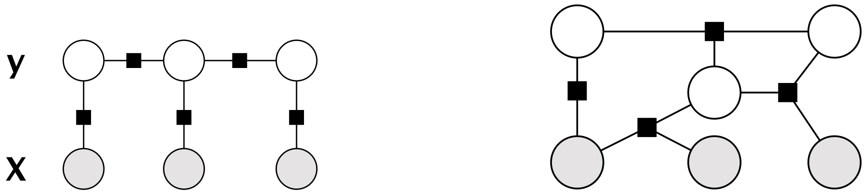
Examples of structured prediction are tasks such as gene segmentation [BCHP07], Named Entity Recognition [ML03] etc.

In structured prediction we have the following setup. For a set of instances $X = \langle X_0, X_1, \ldots, X_k \rangle$ which have the corresponding sequence of labels $\mathbf{y} = \langle y_0, y_1, \ldots y_k \rangle$, where $\mathbf{y}$ is a discrete set of values.

The task is to predict for X the vector of labels $\mathbf{y}$. To do this in an accurate manner, we need to be able to encode the dependencies between instances in the sequence X. Hence, this gives rise to *graphical models* which can encode such inter-dependencies between instances in a sequence.

Conditional Random Fields – CRF. One of the most well known graphical models, for structured prediction, *conditional random fields*, was proposed by Lafferty and McCallum [LMP01]. CRFs belong to the so called *discriminative models* where for a given input sequence X and their labels $\mathbf{y}$, the goal is to model the conditional probability $p(\mathbf{y}|X)$. Here, similarly as in other classification approaches, the input instances are represented in a k–dimensional feature space. The advantage of CRFs and other structured prediction models is that it allows for arbitrary dependencies between instances and correspondingly the labels associated with them.

For this purpose, CRFs represent a sequence as a graphical model, where we distinguish between the *linear-chain* and *general* CRFs. In *linear-chain* CRFs the dependencies between instances in a sequence are limited to immediate neighbors as shown in Figure 2.4a. In *general* CRFs, the dependencies can be arbitrary and are subject to the problem which we try to model.



(a) Linear-Chain CRF

(b) General CRF

Figure 2.4. Modeling of dependencies between instances in a sequence according to linear-chain CRFs and general CRFs.

Figure 2.4 shows the two cases of CRFs. The models are represented as by the graph $G = (X, F, \mathbf{y})$, which is also referred as a *factor graph*. This is because of the nodes F in *black squares* which represent the factors. Through factors in CRFs, we represent the dependencies between the instances in a sequence and their dependency w.r.t the labels. As it can be seen from Figure 2.4b such factors nodes can be arbitrary.

The factors in G are represented by a family of functions $F = \{\Psi_a\}$, which vary in the two models. We first show the estimation of $p(\mathbf{y}|X)$ for linear-chains and then discuss the difference with the general CRFs.

In linear-chains we estimate $p(\mathbf{y}|X)$ as shown in Equation 2.13. The factors Ψ_a here are between two labels y_i and y_{i-1} and correspondingly between the observed sequence X_i and y_i. Hence, the features $f_k(y_i, y_{i-1}, X_i)$ are computed only between the three nodes in G. Here we distinguish, between $f_{ij}(y, y', X)$ which represents features modeling the transition between two consecutive labels y_i and y_j. Next, we distinguish $f_{io}(y, y', X)$ which models the dependency between the observed value X w.r.t the label y.

$$p(\mathbf{y}|X) = \frac{1}{Z(X)} \prod_{i=1}^{n} \exp\left\{\sum_{k=1}^{K} \theta_k f_k(y_i, y_{i-1}, X_i)\right\} \tag{2.13}$$

where $Z(X)$ is a normalization factor which allows us to turn the computed estimates into a probability distribution. θ_k are parameters which indicate the importance of certain features, and they are usually estimated through standard parameter estimation methods like the Maximum Likelihood or other more sophisticated approaches like Quasi-Newton Optimizers [Bis06].

$$Z(X) = \sum_{y \in \mathbf{y}} \prod_{i=1}^{n} \exp\left\{\sum_{k=1}^{K} \theta_k f_k(y_i, y_{i-1}, X_i)\right\} \tag{2.14}$$

Finally, the only difference between linear-chain and general CRFs is in the factor nodes. Since we can have arbitrary dependencies between the input sequences, labels and the factors, instead of $\prod_{i=1}^{n}$ which simply follows the linear-chain, here we have the set of factors Ψ_a in G as shown in Equation 2.15. Figure 2.4b shows that the set of factors is arbitrary and can be between different sequences (in linear-chain a sequence was only factored w.r.t the label, and correspondingly the label was factor w.r.t the previous label) and label nodes in G.

$$p(\mathbf{y}|X) = \frac{1}{Z(X)} \prod_{\Psi_a \in G} \exp\left\{\sum_{k=1}^{K(a)} \theta_k f_k(y_i, y_{i-1}, X_i)\right\} \tag{2.15}$$

2.5.3 Feature Selection

An important aspect that is considered in in supervised learning is *feature selection*. The importance of feature selection is manifold. Through feature selection we choose features that have higher discriminative power on distinguishing the instances of different classes, and further improve on efficiency of the generated classifiers. Furthermore, we can improve the generalizability of a trained model on unseen examples.

Information Gain. We explain a standard feature selection algorithm that is based on the concept of mutual information gain [YP97].

The intuition behind IG feature selection is to assign higher weight to features, respectively to feature values assigned to instances, which enable us to distinguish them into different classes. Table 2.3 shows an example behind the intuition of IG feature selection. It is evident that the example features have high discriminatory power to distinguish between instances belonging to the different classes.

word frequency	*class=winter*	*class=summer*
snow	10	0
ski	25	0
warm	2	50
beach	10	120

Table 2.3. An example feature of word occurrence for classifying documents into *winter* and *summer* class.

To quantify the discriminative or classification power of a specific feature, respectively its values, we measure based on Equation 2.16.

$$IG(f) = H(Y) - H(Y|f) = -\sum_{y \in \mathbf{y}} P(y) \log P(y) + \sum_{k=1}^{n} \sum_{y \in \mathbf{y}} P(y|f^k) \log P(y|f^k) \quad (2.16)$$

We see from $IG(f)$ that through the information gain we simply measure how well we can split based on the feature values instances belonging to the different classes (the right most part of the equation).

3 News in Wikipedia

In this chapter, we analyze how news media, respectively, news articles reporting about certain Wikipedia entity pages are reflected in Wikipedia. As we will show later on, news represent the second most cited source in Wikipedia. Furthermore, based on Wikipedia editing policies [pol], news fall into the category of suggested source types as an authoritative source for citation.

Before delving into details of the main contributions of this thesis, we first analyze the importance of news in Wikipedia. In a controlled study we see if there is a need for automated approaches on suggesting news articles for Wikipedia articles.

The implications of this study are manifold and can be leveraged as following. For instance, automated knowledge base construction tasks can rely on news as a source or an indicator to add or update entities. First, news could be a primary source for addition of emerging entities [HAW14]. Secondly, knowledge bases that build upon Wikipedia can periodically refresh their contents. They constantly deal with the natural trade-off between the maintenance costs of a fresh and consistent state with the loss of useful information. For newsworthy entities and events, understanding this delay in appearing in Wikipedia would suitably help knowledge bases improve their maintenance or characterize the information loss.

We study how fast Wikipedia reacts to real world events reported in news. We carry out this study on the Wikipedia revision history and the New York Times news corpus for the overlapping years between 2001 and 2007. We analyze and define *lag* as the time difference between when an entity or event was reported in news and the first time it appeared in Wikipedia. Specifically, we answer the following questions:

- What fraction of external references in entity pages are news articles?
- How much does Wikipedia lag behind news articles and how does it evolve?
- Which categories or classes of entities in news lead or lag Wikipedia?
- How do events reported by news articles lag with the Wikipedia event pages?

3.1 Collection Alignment

To carry out this study, we first align the two collections, Wikipedia and NYT corpus. The detailed descriptions of the datasets in our experimental setup are given below:

- **Wikipedia** – The *English Wikipedia revision history* [Enc] contains the full edit history from January 2001 to December 2013. We consider all versions including versions that were marked as minor edits.
- **News** – The *New York Times Annotated corpus* [nyt] comprises of more than 1.8 million articles from the New York Times published between 1987 and 2007. Every article has an associated publication time and we refer to this as the time of the article. Since Wikipedia was released in 2001 and the NYT corpus is valid until 2007, we consider the sub-collections from both corpora that are overlapping in time, between 2001 and 2007.

3.1.1 Preliminaries and Setup

Preliminaries. Before delving into detail in the lag analysis, it is necessary to introduce the entity and event notions.

Entity. *An entity is something which has a canonical (i.e., uniquely identifiable) representation in Wikipedia. In other words, it represents a real world concept, e.g.* `People, Organization, Location`. *We refer to the Wikipedia page dedicated to a given entity as an Entity Page.*

Event. *It is defined as a real-world event that has a Wikipedia article, e.g.* `U.S Elections 2004`. *The Wikipedia article dedicated to the event is referred to as the Event Page.*

Setup. The experimental setup is as following. We first link the free text mentions of entities from the news articles in NYT corpus to Wikipedia through *entity linking* (see Section 2.2), for which we rely on TagMe! [FS12]. To maintain high accuracy of the disambiguated entities, we filter out entities with a low threshold[1]. In total, we analyze 1.8 million NYT articles, resulting in approximately 506,151 distinct entities (after filtering for the appropriate threshold). Figure 3.1 shows the distribution of extracted entities for the years 2001–2007, alongside the number of entities appearing in Wikipedia at the respective years.

The final set of entities for our experimental analysis comprises of a collection of 180,478 entities that appear only in the years 2001-2007.

[1] Entities with a disambiguation score lower than 0.3 are filtered out.

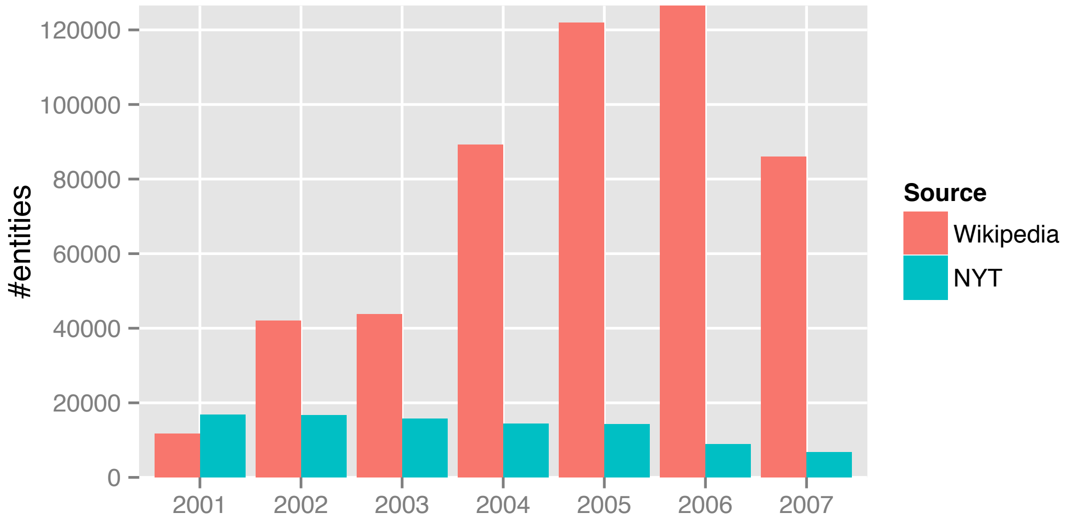


Figure 3.1. Number of entities appearing in the corresponding years in Wikipedia, and those extracted from the entity linking process in the NYT corpus.

3.2 News Reference Density in Wikipedia

To start off we want to investigate how news impacts Wikipedia by studying such news references in *entity pages*. We categorize entity pages based on their types that are associated in DBpedia[2], through the triples $\langle e$ `rdf:type` $type\rangle$. Entity pages, typically contain references to qualify the stated facts therein. These references are broadly classified into the following source types – `web, news, book, report` and `journal`, etc., by Wikipedia[3]. We first study the distribution of news references(of type `news`) in entity pages across *entity types* and define it below.

News Reference Density. *News Reference Density (NRD) of an entity is the fraction of news references over all references of all types in the page. Similarly reference densities of other citation types are defined.*

We observe that, as expected, most of the references are from the `web`. However, the second most dominant type of reference are news references constituting 20% of overall references. The NRD varies across entity categories as shown in Figure 3.2. While types `OfficeHolders` (mostly politicians) have a high news density, on the other hand `Bands` have high density for web references. The NRD in most cases is stable across years for the different entity types as shown in Figure 3.3.

[2] http://wiki.dbpedia.org/Ontology
[3] http://en.wikipedia.org/wiki/Wikipedia:Citation_templates

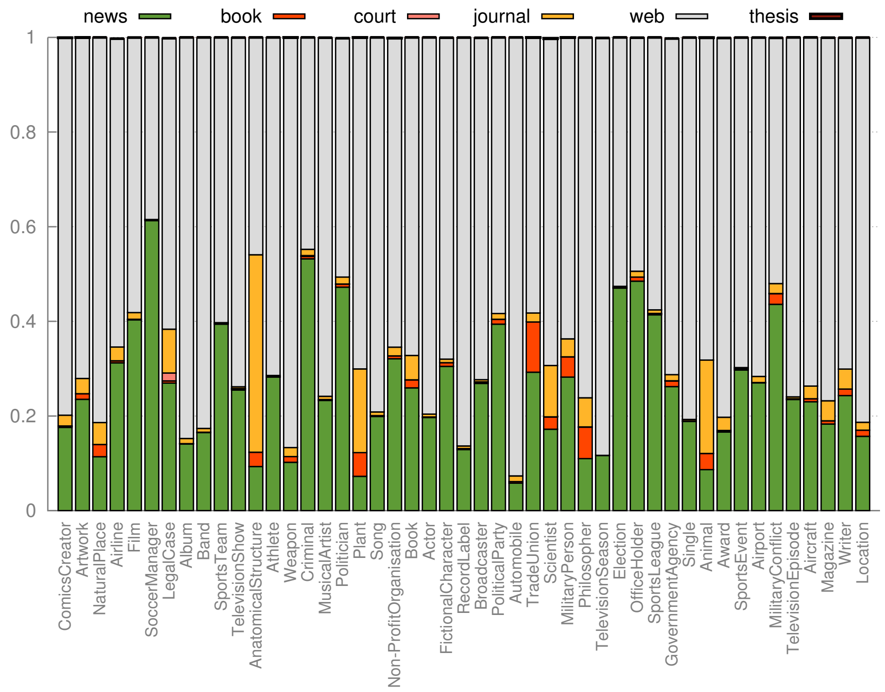


Figure 3.2. News Reference Density for the different entity types. The reference density of a given reference type is measured as the fraction of references of that type over all references for the entity page.

However, there are slight variations on the reference density for specialized types and the corresponding reference types, e.g. `LegalCase` and `Court` reference types.

Taking into account the organization of Wikipedia entity pages into section, we analyze the distribution of news densities across sections in an Wikipedia entities. We observe that sections in entity pages vary considerably across categories with only some of the sections being common among categories, e.g. '*Early Life*' and '*Career*'. When we look at the partial contribution of the sections to the page news reference density, we observe that while '*Early Life and Career*' in `Politicians` have highest NRD contribution of 64%, the section '*Sports Team*' in `Athletes` has the highest contribution of 19%.

3.3 Entity Lag

From the aligned collection we analyze the behavior with which entities extracted from NYT are added in Wikipedia.

Entity Lag. This can be attributed to two factors: *inherent popularity of the entity*, and *evolution of authorship* of entity pages in Wikipedia. One explanation is that

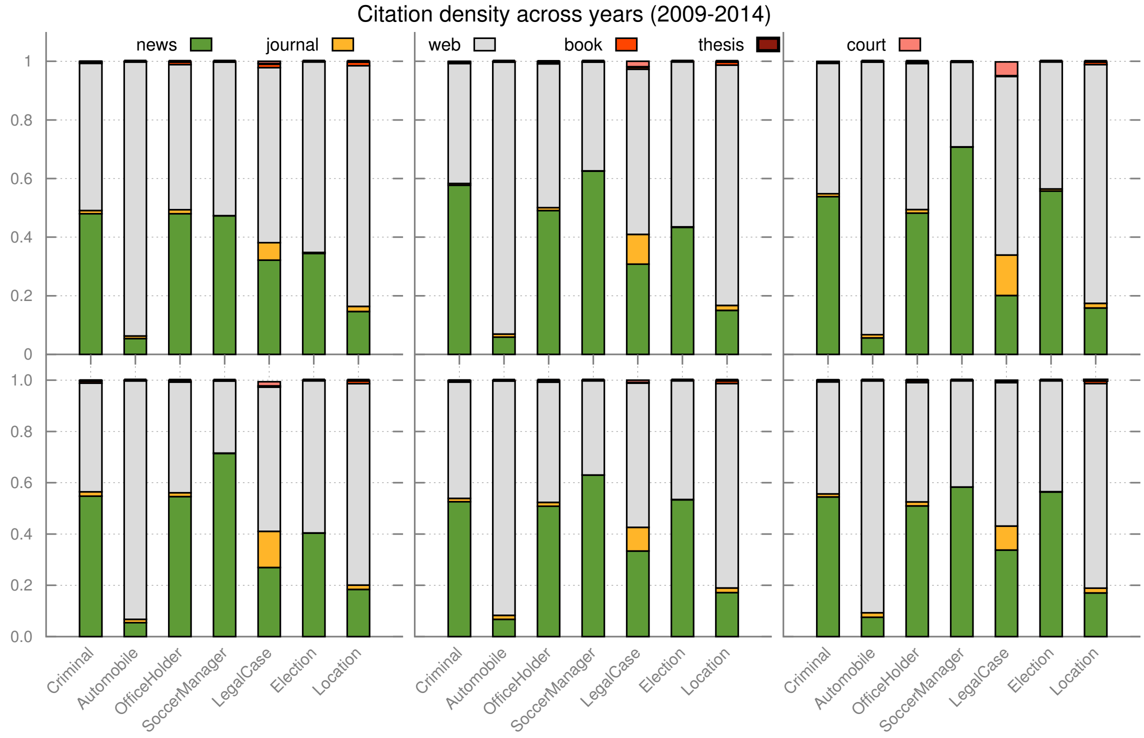


Figure 3.3. Reference density for the different entity types. The plots show the reference density for years 2009-2014, in order from left to right.

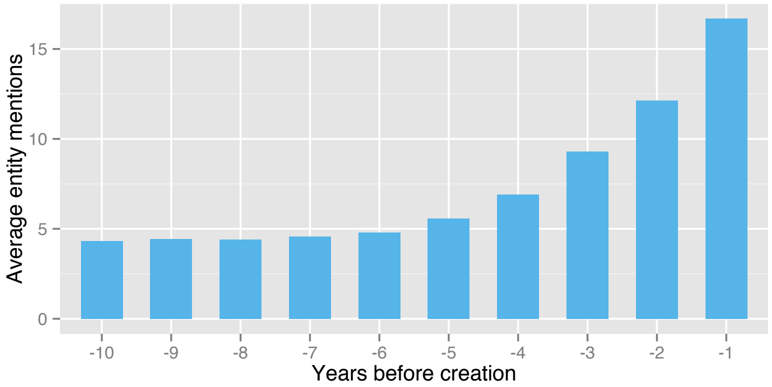


Figure 3.4. Entity mention counts in news articles before creation of Wikipedia entity page. Mention counts of entities peak a year before it is created in Wikipedia.

entities appearing in authoritative news sources like NYT reflect their popularity. Figure 3.4 shows the average entity mention distribution (in NYT) across years before

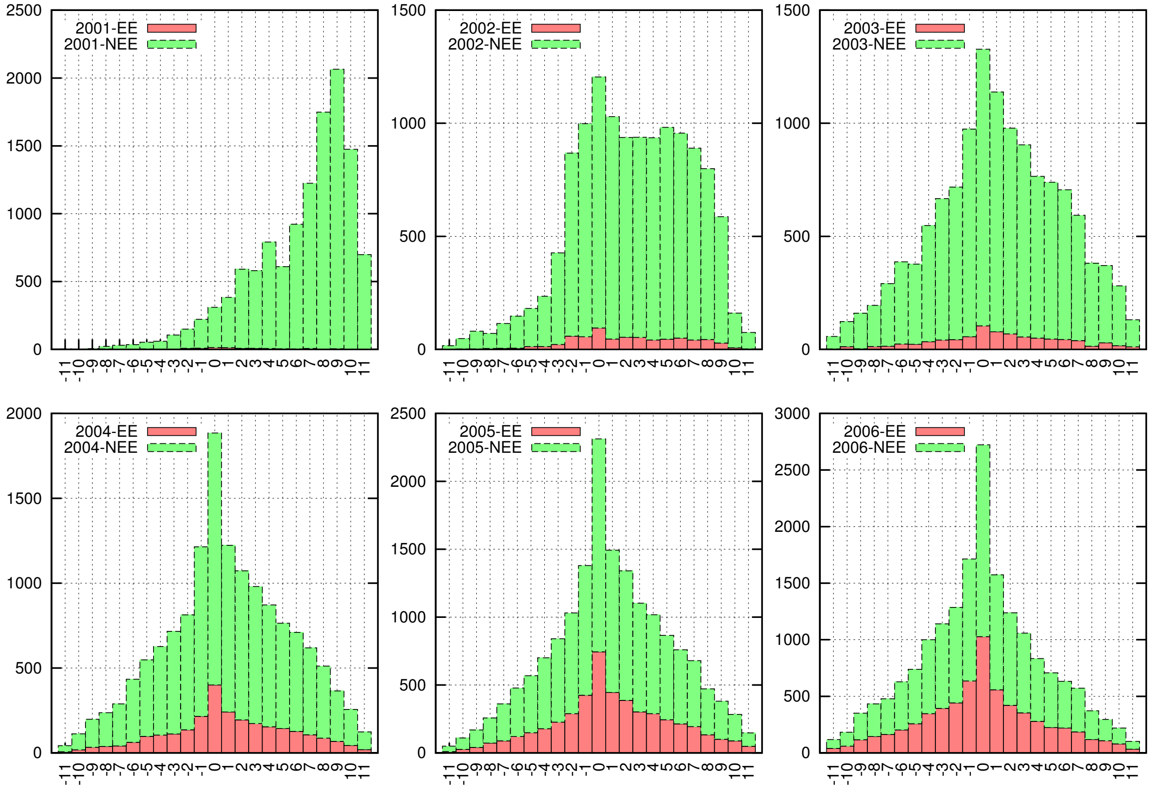


Figure 3.5. Entity lag in months. The emergent entities are shown in red, they are determined by filtering all entities from the subset of NYT that appear in earlier years before 2001. The y-axis is normalized using the *sum* of entities having medium lag for the emerging and non-emerging entities, respectively.

the entity is first created in Wikipedia. This follows the assumption that an increase of entity mentions in news sources will eventually result in the creation of an entity in Wikipedia. Figure 3.4 shows that shortly before the entity creation in Wikipedia, the entity is mentioned most in news. The second factor, is that Wikipedia's authorship has increased with an ever growing number of editors, hence establishing itself as a independent source of information [KGC11], thus entities can be created from what is deemed as important by the editors in Wikipedia.

We measure the *time span* between the entity mention and its creation time in Wikipedia, and define the *entity lag* below.

Entity Lag. *The delay of the first appearance of an entity page relative to the first appearance of its mention in a news article is called* entity lag *or simply* lag $lag(e_i)$. $\mathbf{lag(e_i) = t_w(e_i) - t_n(e_i)}$, *where* $t_w(e_i)$ *is the time of the first version of entity* e_i *was authored and* $t_n(e_i)$ *is the publication time of its first mention in news.*

First, we analyze how the creation of entities in Wikipedia lag their mentions in news. We denote the entities with an absolute lag of less than a month as *low lag entities*, less than a year as *medium-lag entities* and the rest with more than a year as *high-lag entities*. Figure 3.5 shows the lag distribution in months.

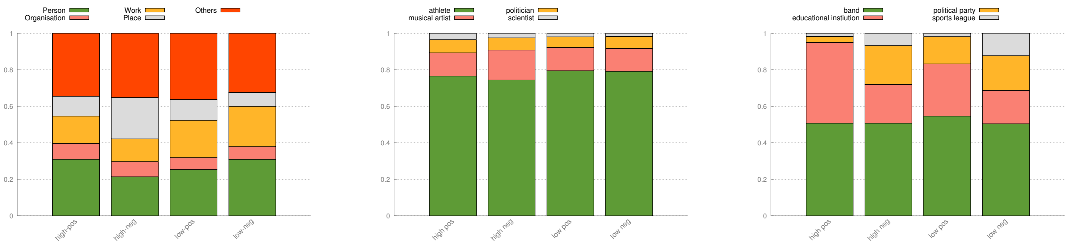
We see that in the first year of Wikipedia the average lag was high with a majority of entities in Wikipedia lagging behind news. However, quite distinctly, the lag redistributes towards a means of zero in the course of time into a Gaussian or normal distribution. We also see that the absolute number of entities with a lag of zeros go up, and the standard deviation reduces. The lag distribution through the years shifts to a normal distribution, with most of the entities centered around the mean, which in our case is zero.

Since Wikipedia only started after 2001, we also consider the entities which were *emergent* in news after 2001 (denoted by the red histogram).

Emerging Entities. *An entity is considered as an* emergent entity *(EE) if its first mention in NYT is after the time when Wikipedia was released, i.e., January 2001.*

Emergent entities have a similar distribution like the existing entities. Since news articles are rich in political news and their coverage, we observe that emergent political topics and entities show low lag. An example is `Freedom Fries` which came into prominence in 2003 as a political euphemism for the actual French fries. On the other hand works of fiction like `The lost City` typically exhibit high lag. Similar to the non-emergent entities the lag distribution for *emergent entities* is normal.

Based on this distributions we can provide rough estimates of the fraction of *'newsworthy'* entities, which could be missed given a maintenance period. Services that periodically update their entity repositories would lose around half of the entities if their update periods is greater than a month than if they update daily.



(a) Overall **(b)** Person **(c)** Organization

Figure 3.6. Lag distribution of different types. The y-axis values are normalized by the sum of the overall entities falling into the different *lag classes*.

3.3.1 Lag for Entity Types

To characterize which entity types show different lag behavior – positive or negative, low or high – we group entities based on their types. For example, *Barack Obama* `isA US President isA Politician isA Person`.

We take a coarse grained representation of entity types, that is *Person, Work, Organization, Places, Other* and are presented in Figure 3.6a. High-positive refers to high lag (Wikipedia lags news) whereas high-negative implies a high lead (Wikipedia leads news). It is natural to see that *Places* have the highest negative lag since entity pages for many geographic locations were introduced during the early days of Wikipedia which we refer to as its *bootstrapping period*.

We see that Wikipedia has a high positive lag for *Persons* (almost 37%) in comparison to other types. This means that most of the emergent entities are people rather than other entity types. A closer look into the four major subcategories of *People* in Figure 3.6b reveals that *musicians* tend to be mentioned in Wikipedia earlier than news and we confirm that most of them, like the locations, were also created during the bootstrapping period. In the case of *Organizations* in Figure 3.8b, we make two observations. First, all educational institutions have a high lag and secondly political parties either have a high lead or a small lag. This suggests that political parties are quite popular entities in Wikipedia while educational institutes are not.

The entity class *Work* encompasses all types of *books, musical composition and movies*. In general *Work* is reported under low lag (around 21%-22%) as compared to its higher lag instances which is around around 12%-14%. In sum, artistic works and locations get reflected in Wikipedia sooner than other types while Wikipedia lags news for emerging personalities. The overall distribution of entity lag is shown in Table 3.1.

lag type	negative (lag)	positive (lead)
high	57.1%	8%
medium	22.2%	11%
low	0.2%	1.1%

Table 3.1. Absolute entity lag distributions for all lag types. The numbers are aggregated over the years 2001-2006.

3.4 Event Lag

We define *event lag*, similarly to *entity lag*, as the publication time difference between the first news article which reports the event and the Wikipedia event page. Events reported in the news can be as a reaction to an event in the past, or a build up to an upcoming event. We do not make a difference in both these cases and treat the first news article reporting the event as the inception of the event.

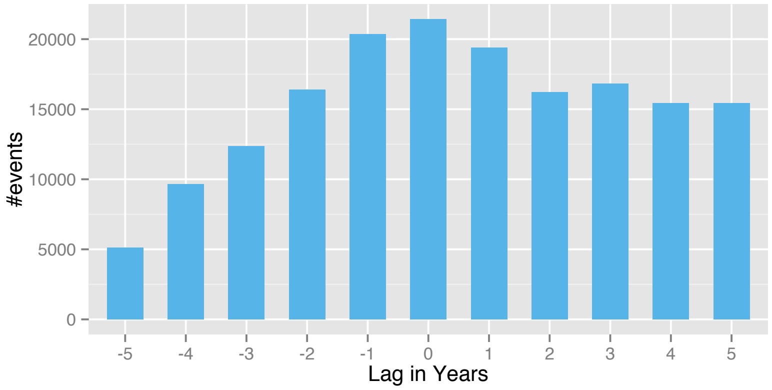


Figure 3.7. Event news reference lag (in years) in Wikipedia. Most of Wikipedia events fall into *low-lag* class, showing high dynamics of reporting real news events in Wikipedia.

3.4.1 Emerging Entities in Event Pages

Finally, we study how events influence the creation of entities in Wikipedia. For this experiment we consider all events in DBpedia with their publication time (resource of type `dbpedia-owl:Event`). Unlike the previous experiments we do not rely on the NYT corpora and hence can consider the entire Wikipedia revision history.

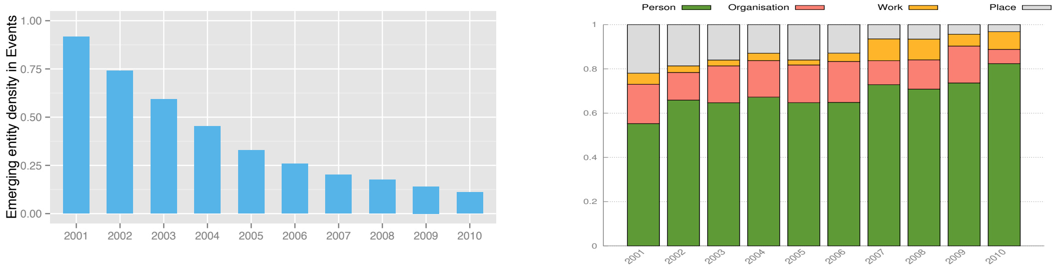
The notion of the publication time corresponds to the first time the event page was introduced in Wikipedia. Next, we extract the explicitly linked entities in the event page and compare the publication times of the mentioned entities and the event publication time. To this effect, we make a simplistic assumption about the mentioned entities in the event page: *entities created after the event page are created because of this event.*

Emerging Entities in Event Pages. *Emerging entity density of an event page is the fraction of entities created after the event page. We refer to them as emerging entities (note that this is different from the emergent entities in the previous section).*

For example, for the event "*Charlie Hebdo Shootings*"[4], created on 7th January, 2015, the mentioned entities therein, "*Corinne Rey*" or "*Coco*"[5] were created five days later on 12th January.

The emerging entity density (EED) evolution from 2001-2010 is presented in Figure 3.8a where the y-axis represents the average emerging entity density of event pages in a given year. We have a total of 14,604 events with 179,981 entities with the exception of events from the last few years owing to the lack of event data in DBPedia for this period. We see that in the early years the EED of event pages was

[4] http://en.wikipedia.org/wiki/Charlie_Hebdo_shooting
[5] http://en.wikipedia.org/wiki/Coco_(cartoonist)



(a) Emerging Entity Density

(b) Emerging entity categories

Figure 3.8. Emerging entity density in Wikipedia event pages.

very high, sometimes above 80%, meaning most of the entities mentioned in the event pages were emerging. Understandably, this declines every year resembling the phenomena of diminishing returns. However, we still see a high percentage of emerging entities in the recent event pages which point to the fact that event pages are great repositories of upcoming and emerging entities missing in the knowledge bases. We also observe that the curve, although decreasing, tends to stabilize in the recent years around 13%. Finally, we look at the categories of emerging entities in Figure 3.6b to find that people comprise the majority of the emergent entities consistently over the years. On the other hand, organizations were emergent between 2001-2005 but their EED contribution to event pages has been decreasing from 2006 onwards.

3.5 Conclusions and Implications

The implications of this study are manifold. First, it shows that news collections are an important resource for mining emerging entities. The normal distribution of the entity lag shows that almost 50% of the entities before occurring in Wikipedia are already mentioned in news. Our experiments on news reference density show that a high proportion of facts about entities are qualified with a news reference.

Secondly, entity and event repositories relying on Wikipedia can quantify the degree of loss or re-calibrate their update frequency based on the lag distribution. It is also possible to optimize emergent entity coverage by focusing on event pages. Interestingly, the lag for events is far lower than entity pages.

Thirdly, event pages are containers of emergent entities with around 12% of the entities linked to an entity being emergent.

Finally, this study provides useful insights and highlights the need for automated approaches to enrich Wikipedia entities and events with news citations. This is one of the key motivations for the contributions in this thesis in the upcoming chapters.

4 Related Work

In this chapter, we review related literature, which focuses towards problems that we highlighted in Chapter 1 and compares the contributions of this thesis.

In more details, in Section 4.1 we review related work which analyzes the dynamics of Wikipedia editors, and the community structures within. This is related to our study in Chapter 3, where we measure the entity and event lag in news and Wikipedia. Next, in Section 4.2, we provide an overview of existing work and state of the art approaches on generating automatically Wikipedia articles, thus, related to our contributions in Chapter 5 and 7. Similarly, in Section 4.3 and 4.4 we list the shortcomings of existing knowledge base acceleration approaches and works on determining entity salience in news articles, as such these represent highly related works relevant for Chapter 5 and 7. Finally, in Section 4.5, we present an overview of existing work in determining the citation span in scientific articles, which is related to our contribution in Chapter 6.

4.1 Wikipedia Editor Dynamics

One of the core parts of Wikipedia are its editors. Therefore, analyzing the behavior of editors and how they collaborate has many implications. First, since Wikipedia is collaboratively created, and thus the guidelines on high-quality edits and other policies are agreed collectively. Second, the analysis on the responsiveness of editors to real-world events and correspondingly how they diffuse such information into the Wikipedia pages is highly important. Our study in Chapter 3 answers one of the questions on how fast are Wikipedia editors to add information reported in news about Wikipedia articles. Below we review work that focuses solely on the Wikipedia editors and the structure of editors therein.

Kittur et al. [KPSM07] analyses the structure of Wikipedia editors on how they collaborate. They further classify the *collaborators* into five different classes based on the number of revisions. Next, they measure the population growth of the collaborators falling into the five different classes. They find that the shift on how content is provided mostly by collaborators with lower number of edits is due to the increased fraction of such users in the Wikipedia community structure. This, however, does not correlate with any decline of the content provided by collaborators with high number of edits, hence, is accounted to the higher fraction of low edit users. In contrast to the work from Kittur et al., we have a different focus in our analysis, namely that of entity and event lag in Wikipedia, without any distinction of the Wikipedia community structure. In [SCCP09] the authors analyze several aspects of Wikipedia's editors. They conclude that the number of edits is decreasing. Another slightly related work [BIA14] analyzes the number of research papers about Wikipedia, here too they conclude that the number has been decreasing, however, papers that use Wikpedia's data has seen an increase.

Keegan et al. [KGC11, KGC12, HLS+07] focus on the dynamics of Wikipedia's coverage of real world entities. In [KGC11], the authors consider emerging events like the Tōhoku catastrophe[1]. In the case of such high dynamic events, it is found out that for localized Wikipedias (e.g. Japanese), the corresponding event appears only six minutes after the event, whereas in the English Wikipedia, it appears in less than an hour. Furthermore, they analyze the co-authorship, concluding that within Wikipedia there are sub-communities that edit articles of the same topic.

As a continuation of their work, in [KGC12] the social network structure of Wikipedia collaborators is analyzed. The analysis is based on four main hypotheses that are based on two main set of attributes, article and editor attributes, respectively. The first hypothesis validates the fact that for breaking news articles attract more editors. The second hypothesis validates the co-authorship of articles in Wikipedia from collaborators that are categorized into three main classes: *Experienced, Apprentice, Non-Expert*. Significant collaborations between the three classes of collaborators is found only on *contemporary* articles (articles are divided into *breaking, contemporary, historical*) between *apprentice* and *experienced* collaborators. The third hypothesis, analyzes the editor attributes and implies that experienced editors will edit more articles than others. The third hypothesis leads to the fourth and last hypothesis. It analyzes the fact that experienced editors are more likely to contribute to similar types of articles rather than to dissimilar. Strong correlation is found for editors belonging to the *apprentice* class and for most of the article types.

The works [KGC11, KGC12, HLS+07, KPSM07] are related only to our study shown in Chapter 3. Contrary to our analysis the work by Keegan et al. has as a main focus modeling the network structure of editors and how this reflects on the dynamics of Wikipedia and contemporary and emergent entities and events. On the other hand, in our analysis we focus on larger real world news corpus which inherently represents emerging entities and events. In addition, we also distinguished the lag of different entity types, and as a last diverging point we analyzed how entities are co-created and their impact on the entity lag.

[1] http://en.wikipedia.org/wiki/2011_T%C5%8Dhoku_earthquake_and_tsunami

4.2 Wikipedia Page Generation

Citation Sources. Ford et al. [FSMM13] analyze the citation behavior of Wikipedia editors with respect to their adherence to the citation guidelines. They investigate what types of sources are most often cited, i.e. *primary, secondary* and *tertiary* as defined in Wikipedia [pol]. Similar as to our study in Chapter 3, they conclude that news are one of the top cited source in the *secondary* type, while they see a growing trend of *primary* sources due to their persistence on the web, contrary to the Wikipedia policies of preferring secondary sources.

Luyt and Tan [LT10], for a subset of Wikipedia entity pages from the domain of *History*, they analyze how citations are biased towards a specific group of sources. [FSMM13, LT10] emphasize the importance of citations in Wikipedia as a means to ensure the quality of entity pages.

Hunag et al. [HWMG14, HKC+12] have considered the problem recommending citations in scientific publications. Despite the fact that this is significantly different from the approaches we propose in Chapter 5, it reveals the importance on the *verifiability* of statement be it in scholarly articles or Wikipedia.

The works regarding citation sources reveal that citation policies in Wikipedia, specifically, the use of third-party sources is important to ensure the high quality of Wikipedia articles, and at the same time, finding citations is an important problem that we encounter in other domains such as scientific literature. Therefore, through our contributions in this thesis, are highly important in other domains as well, apart from our work that focuses only in Wikipedia articles.

Wikipedia Enrichment. Sauper and Barzilay [SB09] propose an approach for automatically generating entire Wikipedia entity pages for specific entity types. The approach is trained on already-populated entity pages of a given type (e.g. '*Diseases*') by learning templates about the entity page structure. For example, an entity of type `Disease`, in majority of the cases they contain a *"Treatment"* section. The approach works as following. For a new entity page, first, they extract documents via Web search using the *entity title* and the *section title* as a query, for example '*Lung Cancer*'+'*Treatment*'. Next, the task is to identify the best paragraphs extracted from the resulting documents from the Web search. They rank the paragraphs via an optimized supervised *perceptron model* for finding the most representative paragraph that is the least similar to paragraphs in other sections. That is, in case an entity has more than one section in its section template, then the paragraph ranking is optimized across all sections such that there is minimal content redundancy from the suggested paragraphs across sections. The top-1 paragraph for each section is added into the entity page.

A similar line of work was proposed by Taneva and Weikum [TW13]. They propose an approach that constructs short summaries for the long tail entities in Wikipedia. The summaries are called '*gems*' and the size of a '*gem*' can be user defined. The size of a summary is measured in terms of characters or words. They focus on generating summaries that are novel and diverse. However, they do not consider any structure of entities, which is present in Wikipedia.

Contrary to [SB09] and [TW13], in this thesis, specifically in Chapter 7, we focus on suggesting entire documents to Wikipedia entity pages, respectively to the appropriate sections. Furthermore, following the Wikipedia editing policies, we focus on news collections as authoritative sources and recommended third-party sources for citation in Wikipedia. The news articles that we suggest for entity pages is where the entity is a *salient* concept in the news article, and that the information contained within the article is novel and is *authoritative* for the entity of interest. The notion of *authority* in this case we explain it subject to the entity of interest. That is, a news article provides authoritative information for an entity of interest if it co-occurs (within the news article content) with entities of higher authority (as measured through prior probability or any centrality measures). This has the advantage as it allows to adjust accordingly on what passes the *suggestion threshold* for an entity page.

The notion of relevance in [SB09] is computed implicitly through the ranking of documents from the Web search. Furthermore, they do not deal with the novelty of a suggested piece of information is proposed to *stub* Wikipedia entity pages.

Finally, through our contribution in Chapter 7 and 5, we address several limitations in this line of work. First of all, the approache [SB09] has the problem of reproducibility and maintainability. This is due to the fact that they rely on Web search which is an uncontrolled variable in this case, and the corresponding ranking is subject to proprietary methodological issues. Secondly, we aim at updating the already existing Wikipedia entity pages, while the focus of related work is on populating *stub* Wikipedia pages. Finally, both the approaches in [SB09] and [TW13] (finding paragraphs and summarization) could be used to process the news articles we suggest to Wikipedia pages. Our concentration on news is also novel.

Wikipedia Quality Measures Anderka et al. [ASL12] propose a supervised approach to predict quality flaws in Wikipedia pages. A quality flaw in Wikipedia is usually annotated with specific *cleanup* tags. They train a model to predict quality flaws, where among the top–10 quality flaws they identify *unreferenced, refimprove, primary sources* as some of the most serious flaws. The work in [ASL12] is complementary to the contributions of this thesis in Chapter 5 and 7. Majority of the quality flaws are due to content in Wikipedia pages not following the editing policies, and one of the major editing policies deals with citations and the source of such citations. Hence, the approaches we propose can address many of the quality flaws.

As part of future work, we foresee combining our approaches together with the one proposed by Anderka et al. [ASL12], and quantify the improvements we gain in terms of quality through our approaches.

4.3 Knowledge Base Acceleration

Cumulative Citation Recommendation (CCR). TREC introduced the CCR track in the Knowledge base acceleration track in 2012. For a stream of news and social media content and a target entity from a knowledge base (Wikipedia), the goal of the task is to generate a score for each document based on how pertinent it is to the input entity. Balog et al. [BRTN13, BR13] propose approaches that find entity mentions in the document collection and rank them according to how central the entity is in the respective documents. This however is a filtering task for documents towards checking if they are relevant for a pre-defined set of entities. In contrast, in our task we aim at finding news citations as evidence for Wikipedia statements.

In the other spectrum of KBA tasks is that of enriching structured knowledge bases. For a specific information extraction template and given corpus, the task is to analyze the corpus and find worthwhile mentions of an entity or snippets that match the templates. West et al. [WGM+14] consider the problem of knowledge base completion, through question answering and complete missing facts in Freebase based on templates, i.e. *Frank_Zappa* `bornIn` *Baltymore, Maryland*. In contrast, in Chapter 7, we do not extract facts for pre-defined templates but rather suggest news articles based on their relevance to an entity. In cases of long-tail entities, we can suggest to add a novel section through our abstraction and generation of section templates at entity class level.

4.4 Entity Salience and Filtering

An important aspect in this thesis, is to determine salient entities on a given corpus. In our thesis, we focus solely on news articles, which have a distinct language style, referred to as the *pyramid style*, where the important information is mentioned first. In this section, we review some of the most prominent works in this field, and compare to our proposed entity salience models in Chapter 7.

Determining which entities are prominent or salient in a given text has a long history in NLP, sparked by the linguistic theory of Centering [WJP98]. Salience has been used in pronoun and co-reference resolution [Ng10], or to predict which entities will be included in an abstract of an article [DG14]. Frequent features to measure salience include the frequency of an entity in a document, positioning of an entity, grammatical function or internal entity structure (POS tags, head nouns etc.). These approaches are not currently aimed at knowledge base generation or Wikipedia coverage extension but we postulate that an entity's salience in a news article is a prerequisite to the news article being relevant enough to be included in an entity page. Therefore, in Chapter 7, as part of our entity salience model, we use the salience features in [DG14]. However, these features are document-internal — we will show that they are not sufficient to predict news inclusion into an entity page and add features of entity authority, news authority and novelty that measure the relations between several entities, between entity and news article as well as between several competing news articles.

As we will show in Chapter 7, our entity salience models which relies on the language style of news significantly outperform existing approaches [DG14].

4.5 Citation Span

The related literature in this section focuses solely on the contributions in Chapter 6. The approaches we propose in determining the span of a citation in Wikipedia articles are novel and as such do not have directly related works in the same domain. However, there exist previous work that consider this problem in the scientific domain of, on determining the scope of a reference in a scientific article. As we will discuss in more details below and later show in our experimental evaluation, there are major differences on how citations are used in Wikipedia and in scientific publications.

Scientific Text. One of the first attempts to determine the citation span in text was carried in the context of document retrieval [O'C82]. Here we encounter the first mention of *citing statements*, which refers to the sentences in a given document that are relevant or describe a reference to another document or scientific article. In this work, the *citing statements* from a document were used as an index to retrieve the *cited* document. The citing statements are extracted based on heuristics starting from the citing sentence and are expanded with sentences in a window of +/-2 sentences, depending if they contain cue words like *'this', 'these',… 'above-mentioned'.* We consider the approach in [O'C82] as a baseline.

Kaplan et al. [KTT16] proposed the task of determining the *citation block* based on a set of *textual coherence* features (e.g. grammatical or lexical coherence). The citation block *starts* from the citing sentence, with succeeding sentences classified if they belong to the block. The classification of succeeding sentences after the citing sentence is performed through supervised models either relying on standard discrete output classifiers like SVMs or sequence prediction through CRFs.

Abu-Jbara et al. [AR12] determine the citation block by first segmenting the sentences and then classifying individual words as being *inside/outside* the citation. Finally, the segment is classified depending on the word labels (majority of words being inside, at least one, or all of them). This approach is not applicable in our case due to the fact that words in Wikipedia text are not domain or genre specific as one expects in scientific text, and as such their classification does not work.

Citations in IR. The importance of determining the citation span has been acknowledged in the field of Information Retrieval (IR). The focus is on building citation indexes [Gar55] and improve the retrieval of scientific articles [RRT08, RTR06].

Indeed, citing statements, are highly valuable as they provide an abstraction or summary of the cited documents and as such can greatly improve the retrieval process of scientific articles. In majority of the cases, the citing statements consist of the citing sentence and sentences extracted from fixed window size.

Citations for Summarization. Citations have been successfully employed to generate summaries of scientific articles [QR08, ESF$^+$08]. In all cases, citing statements are either extracted manually or from heuristics such as extracting only the citing sentences. Similarly [NO99] expands the summaries in addition to the citing sentence based on cue words (e.g. *'In this', 'However'* etc.). The work in [QR10] goes one step beyond and considers sentences which do not *explicitly* cite another article. The task is to assign a binary label to a sentence, indicating whether it contains context for a cited paper. Since the work in [QR10] tackles a similar problem to the one we address in Chapter 6. Similarly, here too the premise is that citations are marked explicitly and additional citing sentences are found dependent from them. We compare against it and show that independent of the methodological differences, that there is a major difference between scientific articles and the language in web and news sources.

The language style and the composition of citations in Wikipedia and in scientific text differ significantly. Citations are *explicit* in scientific text (e.g. *author names*) and are usually the first word in a sentence [AR12]. In Wikipedia, citations are *implicit* and there are no cue words in text which link to the provided citations. For example, for a statement in Wikipedia articles a reference to a web or news source is supposed to provide evidence for the statement therein. Hence, the link between the two is only explicit when we analyze the content of the statement and the reference, that is, there are no explicit cues in the statement that point to the reference. Therefore, the proposed methodologies and features from the scientific domain do not perform optimally in the case of determining the span of a citation in Wikipedia.

In the case of [QR10], we show that despite the similarities in the objectives of their approach, however, due to the differences in domains and the sentence level granularity it leads to erroneous spans in Wikipedia. This similarly holds for [O'C82] where as shown in Table 6.1, in Wikipedia, the citation span needs to be performed at the sub-sentence level.

Related to our problem is the work on addressing quotation attribution. Pareti et al. [POK$^+$13] propose an approach for addressing the *direct* and *indirect* quotation attribution. The task is mostly based on lexical cues and specific *reporting verbs* that are the signal for the majority of direct quotations. However, in the case of quotation attribution the task is to find the *source, cue*, and *content* of the quotation, whereas in our case, for a given citing paragraph and reference we simply assess which text fragment is covered by the reference.

The approach in Chapter 6 is the first attempt to determine the span of citations for *web* or *news* references in Wikipedia. The approach therein can help further improve and enforce the editing policies by explicitly marking the span of a citation in a Wikipedia article.

5 Finding News Citations for Wikipedia Entities

Wikipedia has become the most used Internet encyclopedia, and indeed, one of the most popular websites overall.[1] In addition, due to Wikipedia's inclusion into widely used applications such as Google KnowledgeGraph or Apple's Siri system, its content will influence the knowledge, and potentially the behavior of millions of users, even if they do not visit the Wikipedia site directly. Therefore, it is essential that its content is accurate and reliable.

Contrary to traditional encyclopedias, Wikipedia is not authored mainly by experts. Articles are authored collaboratively by more than just a small number of contributors and the identity and expertise of authors is hard to verify. This leaves Wikipedia articles open to addition of inaccurate content, spamming or vandalism, and calls into question its reliability. A substantial number of reliability studies have compared Wikipedia against other reference works (such as the *Encyclopedia Britannica* or drug package information) or subjected them to expert review: The exhaustive survey in [MOM+15] concludes that the results of these studies have overall been favorable to Wikipedia when it comes to accuracy of facts, although some works (especially on medical articles) found errors of omission.[2]

These surprisingly favorable results on the reliability of Wikipedia can in all probability be traced to a small number of Wikipedia editorial policies [pol], one of which we are concerned with in this chapter. The *Verifiability* policy requires Wikipedia contributors to support their additions with citations from authoritative external sources. In particular, Wikipedia policy states that *"articles should be based on reliable, third-party, published sources with a reputation for fact-checking and accuracy."*[3] This policy, on the one hand, guides contributors towards both neutrality and the importance of authoritative assessment, and on the other hand, allows Wikipedia

[1] In 2017 it was in top–10 most visited sites according to Alexa www.alexa.com.

[2] The standard for medical information should be higher for obvious reasons and omitted information for side effects or risks can be crucial.

[3] https://en.wikipedia.org/wiki/Wikipedia:Identifying_reliable_sources

core editors to identify unreliable articles more easily by the lack of such citations. Citations therefore play a crucial role in ensuring and upholding Wikipedia reliability.

For current and recent events, as we showed in Chapter 3, news citations are one of the most-used sources [FAA15]. Again, Wikipedia encourages the use of news outlets as citations: *"news reporting from well-established news outlets is generally considered to be reliable for statements of fact"*[3]. Indeed, news are the second-most widely used citation category in Wikipedia (with 1.88 million citations in our English Wikipedia snapshot). However, 26% of these are no longer available due to dead or redirected links. In addition, new information is added all the time and will need verification. For both these purposes, an automatic way of finding an authoritative news citation for any fact(s) one might wish to update, locate again or add would greatly facilitate Wikipedia editing and improve its reliability. Moreover, if no such citation can be found, it can guide contributors or core editors towards questioning their edits.

In this chapter, we describe an approach, which deals with the problem of finding automatically news citation for Wikipedia entities. In particular, we make the following contributions:

- We analyze for which type of Wikipedia statements a news citation is appropriate (in contrast to, for example, a scientific journal citation), taking into account the type and structure of entity the statement is about, as well as the language the statement is written in. We provide a supervised learning algorithm for *statement classification* into the different citation categories.
- We develop a citation discovery algorithm which formalizes three properties of a good citation, namely that it *entails* the statement it supports, that it is from an *authoritative* source and that the statement it supports is *central* to it.
- We establish a large-scale evaluation framework for citation discovery which uses crowdsourcing for measuring our approach's precision.

To the best of our knowledge, this is the first work that automatically discovers citations for fine-grained Wikipedia statements. We show that news citations can be discovered with high precision, in large contemporary news collections. In particular, we with high accuracy recover the same or very similar citations as the ones originally given by Wikipedia contributors in the presence of numerous strong distractors or even find citations which are preferable to the original ones (as established via crowdsourcing).

5.1 Problem Definition and Approach Outline

In this section, we describe the terminology and problem definition for finding *news citations* for Wikipedia entities.

5.1.1 Terminology and Problem Definition

We operate on a specific snapshot of Wikipedia $\mathbf{W}$ where the text in each Wikipedia page $e \in \mathbf{W}$ is organized into *sections* denoted by $\Psi(e)$. Additionally, entity pages are organized into a *type* structure, which is a *directed-acyclic-graph* (DAG) induced by the Wikipedia categories.

This is routinely exploited by knowledge bases like YAGO (e.g. *Barack_Obama* `isA Person`) and we leverage this type structure where each page e belongs to a set of types $T(e)$. We, however, modify the original YAGO type structure to make it *depth consistent* as explained in Section 5.3.3.

Citations and Wikipedia Statements

Here, we describe the necessary terminology to explain the proposed approach for finding news citation for Wikipedia entity pages.

- **Citation:** In Wikipedia pages, any *piece of text* can be supported by a *citation*. The citation points to an external information source, such as a news article, blog, book or journal, that is considered as *evidence* for the fact mentioned in the text.

  Citations in Wikipedia are categorized into a predefined set of *16 citation categories*, c = {`web`, `news`, `books`, `journal`, `map`, `comic`, `court`, `press release`, ...}. Figure 5.2 shows the distribution of the different citation types.

- **Statement:** We refer to the *piece of text* from a Wikipedia page, which *has* or *needs* a citation as a *Wikipedia statement* or simply a *statement*.

  We restrict statements to a single sentence or a sequence of sentences that occur between two consecutive citation markers or a citation marker and paragraph beginning/end. A *citation marker* is either an actual citation or a placeholder `citation needed`[4].

  Each statement s in a page e belongs to a section $\psi \in \Psi(e)$, and the set of statements extracted from a section ψ of e is represented as $S(e, \psi)$.

- **Anchors and Entities:** Typically words or phrases in statements link to other Wikipedia pages which represent entities through *anchors*.

  We denote these links to other pages or entities starting from a statement s as $\gamma(s)$, and $T(s) = \{T(e) \,|\, e \in \gamma(s)\}$ the corresponding entity types.

[4] https://en.wikipedia.org/wiki/Template:Citation_needed

Citation Finding Tasks

Here, we present an overview of our approach on finding news citations. We decompose our approach into two main steps, which we describe below.

- **Statement Categorization.** In this task the aim is to determine for a statement s the appropriate citation category c. This step is in particular important as it allows us to consider in the later steps only those statements for which we are in hold of an appropriate corpus, e.g. *news collection*.

  We define the following function, which for an entity page e and a statement s predicts the correct citation category c.

  $$SC : f(s, e) \rightarrow c, \text{where } c \in \{\texttt{web, news,}\ldots\} \tag{5.1}$$

  We want to categorize s as a requiring a *news* citation if a news citation is the most appropriate citation type. This is based on the hypothesis that each statement typically has a preferred citation category, which we need to determine before making a high precision citation recommendation. We call a statement s that requires a news citation as a *news statment*.

- **Citation Discovery.** In this task we are given a statement s, which based on the previous step, we classify its citation category to be $c =$*news*, and a news collection $\mathcal{N}$.

  We define the *citation discovery* as the task of finding news articles $n \in \mathcal{N}'$, where $\mathcal{N}' \subseteq \mathcal{N}$, which can be suggested as a citation for statement s.

  $$FC : f(s, e, \mathcal{N}) \rightarrow \langle s, n \rangle \in \{`correct', `incorrect'\} \tag{5.2}$$

5.1.2 Approach Overview

Figure 5.1 shows an overview of the proposed approach. For an entity, we extract the section and type structure. Additionally, from the individual section we extract its statements and for each statement we run both steps for statement categorization and citation discovery.

1. ***Statement Categorization–SC.*** In this step, we predict the citation category of a Wikipedia statement s via a supervised machine learning model which we train on features we extract from the statement itself and the entity e. We train a multi-class classification model, where the classes correspond to the citation categories c.

2. ***Citation Discovery–FC.*** For all the *news statements* as classified in the previous step, we find evidence via news articles from our news collection $\mathcal{N}$. We retrieve *candidate* news articles from a news collection $\mathcal{N}$ through standard *information retrieval* methods with s serving as our query, and classify each candidate as either an appropriate citation for s or not.

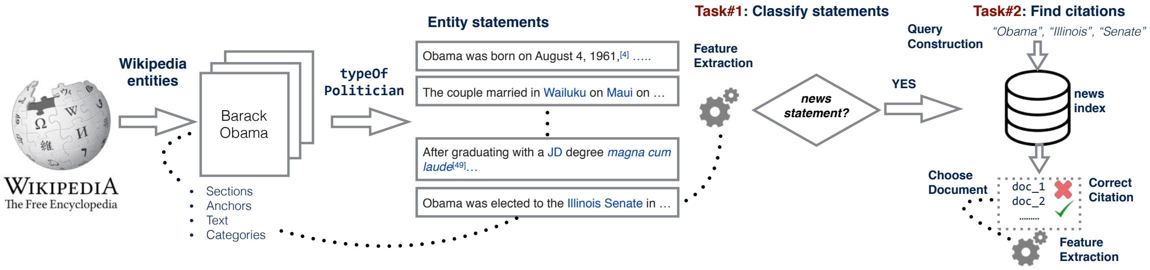


Figure 5.1. Approach overview. In the first task, we classify statements into one of the citation categories, and in the second we find the appropriate news article for citation for a news statement.

5.2 Wikipedia Ground-Truth

Before delving into the details of the proposed approach, we first describe the ground-truth dataset we use to evaluate our approach. We describe two main aspects: (i) the extract of statements from Wikipedia and their corresponding citation categories, and (ii) the news collection which consists of news articles already references from Wikipedia entity pages.

5.2.1 Ground-Truth: Wikipedia News Statements

We extract all statements as defined previously in Section 8.1.1 from a Wikipedia snapshot $\mathbf{W}$[5]. Additionally, for each statement we extract all the citations to the external references. A statement can have more than one citation and in all cases we assume that a citation is valid for the entire statement[6]

In total, we extract 6.9 million statements which point to 8.8 million citations. After analyzing the set of entity pages to which these statement are assigned, we are left with 1.65 million entities and a total of 668 distinct sections.

As described previously, each citation belongs to a category which is chosen by the Wikipedia editor from a citation category template. However, since there are

[5] The snapshot corresponds to the state of Wikipedia accessed at 2015-07-01

[6] As a statement can have different clauses, sometimes extracted citations only serve as evidence for part of the statement. We, however, do not distinguish at this level of granularity but assume that all associated citations support the whole statement.

no policies that enforce a strict categorization of citations, in some cases, Wikipedia editors assign the wrong categories. For instance, in cases where the citation refers to a news article, instead of `news` category, categories like `web` are assigned.

Ground-Truth Curation. For example, from the collection of citations in **W**, the top–3 news domains *BBC*, *NYTimes*, *Guardian*, are often cited in categories other than `news`.[7] Most of such *violations* by the editors occur when citing news under the category `web`. Indeed, the citation category `web` has a wide scope of its validity, however, in cases where a more fine-grained category can be chosen than the use of `web` category is inappropriate.

In most cases such violations can be accurately corrected by applying simple heuristics. We propose two simple, yet effective heuristics for this purpose:

- **Majority Voting.** Citations from the same domain URL are tagged with different categories. We resolve such cases based on *majority voting*. In case a domain is cited more often under the `news` category, then all citations to the same domain are changed to `news`.
- **URL Patterns.** In this heuristic we look for patterns in the URL, specifically for '*/news/*' and '*http://news.*'. This rule is applied to `web` statements, and in case the URL matches one of the patterns, we change its category to `news`.

Table 5.1 shows the top–4 most frequent citation categories and the impact of our ground-truth curation rules. Rule application changes the citation category for 1.65 million citations, approximately 18% of all citations in **W**. The cells in the table show the number of statements that are changed from the category in the *row* to the category in the *column* table.

	book	journal	news	web
book	0	2,650	1,155	71,801
journal	14,905	0	13542	110,133
news	5,698	2,770	0	**391,634**
web	16,549	25,109	**944,977**	0

Table 5.1. The cells show the number of statements that are changed from one category to another category after ground-truth curation.

Finally, we assume that a statment s is a *news statement* if it contains at least one news citation (after the ground-truth curation). Figure 5.2 shows the statement

[7] Thus, the citation http://news.bbc.co.uk/1/hi/uk_politics/7433479.stm from the entity `Liam Byrne` has been categorized as `web`, although the more specific `news` category would have been appropriate.

distribution across the citation categories. It is evident that `web` and `news` are the two most popular categories, with 5.3 and 1.88 million citations, coming from 1.2 million and 436k entities, respectively.

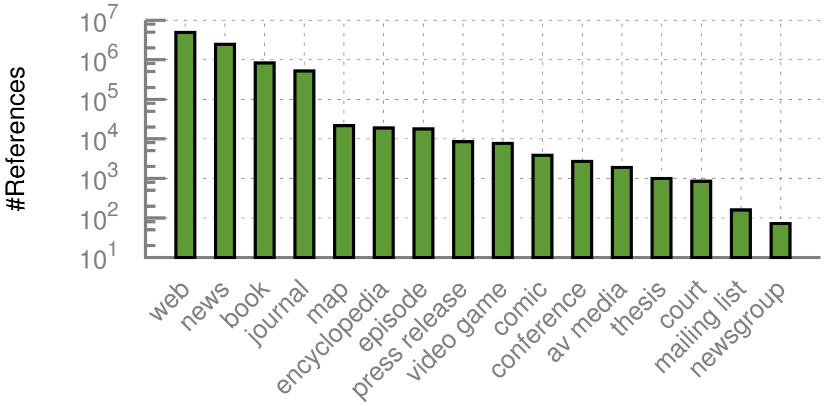


Figure 5.2. Statement distribution by citation category.

5.2.2 Wikipedia News Collection

From the *news statements*, we extract the cited news articles and construct the news collection, which we refer as the *Wikipedia news collection*, $\mathcal{N}^W$. The collection $\mathcal{N}^W$ serves as our ground-truth for the *citation discovery* task.

We distinguish the following sub-collections from $\mathcal{N}^W$. With N_s we denote the set of news articles cited by the news statement s, and with $N_t \subseteq \mathcal{N}^W$ we refer to the set of news articles cited from news statements s whose entities are of type t.

The collection $\mathcal{N}^W$ consists of 1.88 million news articles, from which we successfully crawled 1.55 million articles. The remaining 19% point to non-existent articles (dead links, moved content, etc.). Furthermore, some of the successfully crawled URLs point to the *index pages*. This can be noticed when we consider the article length (in terms of characters) in Figure 5.3. After filtering news articles whose length was below 200 characters, we are left with 1.39 million articles. This presents a decrease of 26% from the original set of 1.88 million news articles.

An additional issue we notice in $\mathcal{N}^W$ are citations to non-English news articles. We find that 23% of articles in $\mathcal{N}^W$ are in languages other than English. We use the Apache Tika[8] for detecting the language of the news articles.

[8] http://tika.apache.org

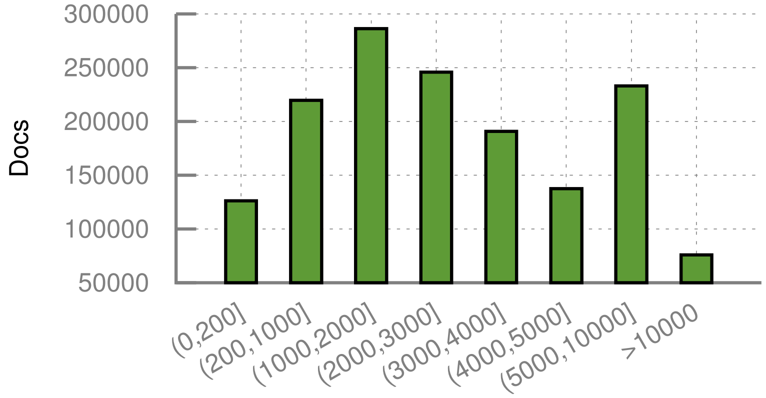


Figure 5.3. News article length (in number of characters) distribution.

5.3 Statement Categorization

In the statement categorization task, we are given a statement s and the entity e from which it is extracted. We compute features that exploit the language style of s and the type and section structure of e to categorize s into one of the citation categories c. We learn a multi-class classifier (Section 5.3.3) with classes corresponding to citation categories c and optimize for predicting news statements.

Table 5.2 shows an overview of the features that we compute for the function SC to categorize a statement s.

feature	description	
$\#verbs\_attr$	the number of verbs of attribution in statement s	*Language Style*
$\#POS$	the frequency of POS tags in s	
$\lambda(s)$	temporal proximity of s to the time-point of $\mathbf{W}$	
$discourse$	explicit discourse annotations of s	
$\#quotations$	the number of quotations in s	
$\theta(s, N_t)$	KL score between the LM of s and that of N_t	
$LDA(s, N_t)$	similarity of s to the topic model computed from N_t	
$p(s = news\vert\psi)$	section and type priors, with *min, max* and *avg* scores	*Entity Structure*
$p(s = news\vert t)$		
$p(s = news\vert t', t)$	type co-occurrence probability between $t \in T(e)$ and all the possible types from entities in s, that is $t' \in T(s)$	
$p(s = news\vert t, \psi)$	type-section probability scores	

Table 5.2. Feature list for statement categorization.

5.3.1 Statement Language-Style

We hypothesize that Wikipedia news statements are similar to the language style of news, as they often paraphrase cited news articles. Different genres (such as *news, recipes, sermons, FAQs, fiction . . .*) differ in their linguistic properties as the different functions they fulfill influence linguistic form [Bib91]. For example, we expect news reports (which center mostly on past events) to contain more past tense verbs than a recipe which gives instructions via verbs in the imperative. We use features that were successful in automatic genre classification including structural features via parts-of-speech as well as lexical surface cues [PW11].

Part of Speech Density. Frequency of part-of-speech (POS) tags, determined via the Stanford tagger, allows us to capture some of the structural properties of text. For example, news statements can be characterized by a high number of past tense verbs as well as proper nouns. We normalize the POS tag frequency w.r.t the sum of all tags in a statement, to account for varying statement length.

Verbs of Attribution and Quotation Marks. News articles often report statements by persons of repute, witnesses or other sources. We approximate this kind of behavior in news articles through two features: Firstly, we count *verbs of attribution* in s, via a list of 92 such verbs (*claim, tell etc*) with POS tag `VB*` and normalize w.r.t the total number of `VB*`. Secondly, we use **quotation marks** as a potential indicator of *paraphrasing*. The feature simply counts the number of quotation marks in s, normalized w.r.t the statement length.

Temporal Proximity $\lambda(s)$. Most Wikipedia ness statements refer to relatively recent events, i.e. events close to the time of the Wikipedia snapshot. We use *temporal expressions* such as dates and years as distinguishing features for news statements. We use a set of hand-crafted regular expression rules to extract temporal expressions.[9]. We use the following rules: (1) `DD Month YYYY`, (2) `DD MM YYYY`, (3) `MM DD YY(YY)`, (4) `YYYY`, with different delimiters (whitespace, '-', '.'). We then compute $\lambda(s) = |Year(\mathbf{W}) - Year(s)|$.

In this case, for temporal proximity we could rely on more accurate temporal cues to measure the distance between the news statement and the revision date of an entity when we first encounter s. However, to do so, we would need to trace all the changes in an entity page to be able and determine the accurate revision date for s. Therefore, we opt for the more simplistic approach where the distance is in terms of years, and is with respect to the year of the Wikipedia snapshot.

[9] This proved to be more scalable than state-of-the-art extractors like HeidelTime [SG10] and Stanford's CoreNLP [FGM05] module

Discourse Analysis. We use discourse connectives to annotate the statements s with *explicit discourse relations* based on an approach proposed by Pitler and Church [PN09]. The annotations belong to the categories {*temporal, contingency, comparison, expansion*}, following the Penn Discourse Treebank annotation [PDL+08]. Some of the explicit discourse relations are particularly interesting (i.e., *temporal*) as they represent a common language construct used in news articles that report event sequences. The features are boolean indicators on whether s contains a specific explicit discourse relation.

Language Model and Topic Model Scoring. As surface lexical features have been shown to be efficient in genre recognition [SWM10], we compute n–gram (up to n=3) language models with *Kneser-Ney* smoothing (LM) from news articles N_t and compute the Kullback-Leibler divergence score between the corresponding language models, $\theta(s, N_t)$. The score shows how likely s can be constructed from the LM from news articles cited from entities of type t.

Similarly, we compute topic models using the LDA framework [BNJ03], where the score is the Jaccard similarity between s and the topic terms from N_t.

5.3.2 Entity-Structure Based Features

Determining if a statement requires a news citation solely on language style is not always feasible. We exploit the entity structure of e and compute the probability of statements having a news citation given its types $T(e)$ and sections $\Psi(e)$.

Section-Type Probability. A good indicator of the likelihood that a statement s requires a news citation is the entity type it belongs to and the section that it appears in. For instance, for type `Politician`, news statements have higher density in section *'Early Life and Career'* as these tend to be more reflected in news.

Since the number of combinations between entity types and sections can be large. One precaution we need to take into account is over-fitting. We avoid over-fitting by filtering out entity types with fewer than 10 statements. Similarly, we filter out sections with fewer than 10 statements, and in which case all the statements belong to the same citation category.

We compute the probability of a statement s having a news citation given the entity type it appears in and the corresponding section in the entity page e.

$$p(s = news|t, \psi) = \frac{\sum_{e \in \mathbf{W} \wedge t \in T(e)} \sum_{s \in S(e,\psi)} 1_{s\ \texttt{typeOf}\ \ \text{news}}}{\sum_{e \in \mathbf{W} \wedge t \in T(e)} |S(e, \psi)|} \tag{5.3}$$

The $p(s = news|t, \psi)$ probability is likely to be a sparse feature, so we compute type and section news-priors. We compute section $p(s = news|\psi)$ and type news-priors $p(s = news|t)$ as the ratio of news statements over the total number of statements (of any citation category) that belong to a section ψ or type t, respectively.

Since s is associated with an entity e, which consequentially may be associated with a set of types $T(e)$, we aggregate the computed type news-priors and the section-type probability into their *min, max* and *avg* scores.

Type Co-Occurrence. From the entity types $T(s)$ and $T(e)$ we measure the likelihood of type co-occurrence in news. The probability simply counts the co-occurrence between t and t' in news statements with respect to their total co-occurrence. Examples of highly co-occurring types in news are `Politician` and `Organization`.

$$p(s = news|t', t) = \frac{\sum_{e\in\mathbf{W}\wedge t\in T(e)} \sum_{s\in S(e)\wedge t'\in T(s)} 1_{s\ \texttt{typeOf}\ \ \text{news}}}{\sum_{e\in\mathbf{W}\wedge t\in T(e)} \sum_{s\in S(e)} 1_{t'\in T(s)}} \tag{5.4}$$

5.3.3 Learning Framework

Learning Setup. Wikipedia consists of a highly diverse set of entities. A model trained on all entities is unlikely to work. For example, the types `Location` and `Politician` represent two highly divergent groups with regard to their page structure, the statements they contain and the way they are reported in news.

Therefore, we learn SC for individual types in the YAGO type taxonomy. The advantages of type specific functions SC is that they are trained on homogeneous entities, which helps the models to predict with higher accuracy. We take only types that have more than 1000 entity instances, resulting in a total of 672 types from the YAGO type taxonomy. The types are organized from very broad types such as (`owl:Thing`) to very specific types like `Serie_A_Players`.

To utilize the specialization and generalization in a principled manner we transform the YAGO type taxonomy (DAG) into a hierarchical DAG. This is utilized later on in order to find the right level of type granularity for learning SC. That is, if we start at the root of the type taxonomy, we will have all possible entities, and as we go down the type taxonomy the number of entities per type decreases, however, the entities are more homogeneous.

We assume that the hierarchy is rooted at `owl:Thing` and all internal nodes are *depth-consistent*, i.e. all paths from the root to the node are of the same length. We obtain this by a simple heuristic whereby for every *child type* → *parent type* we remove edges where the parent's *depth level* in the taxonomy is higher than the *minimum level* from other parent nodes. Figure 5.4 shows an example of how we construct a depth consistent type taxonomy.

With this hierarchical type-taxonomy, we can determine the *optimal level* of type granularity such that we have optimal performance in categorizing statements. For learning the type specific SC, we keep 10% of entity instances for evaluation and

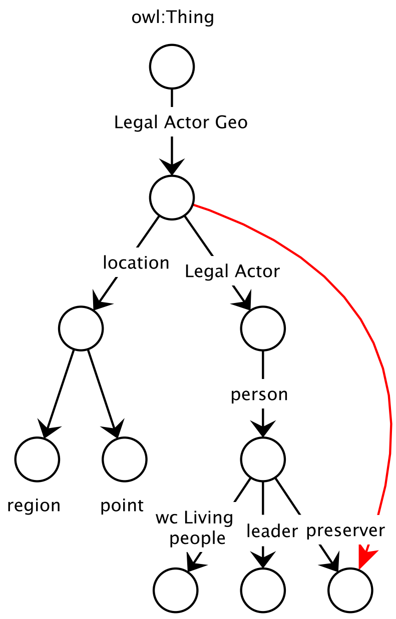


Figure 5.4. An example sub-tree from the YAGO type taxonomy. The edge in red represents the edge we need to delete since it is not depth-consistent.

the remainder for training. It is important to note that when we learn *SC* for a given type, the training instances are sampled through *stratified sampling* from all its *children types*.

Learning Model. The functions *SC* represent *multi-class* classifiers with classes corresponding to the citation categories. Since we want to predict the news category $c =$'*news*' with high accuracy, one question is why we do not pose this as a *binary* classification problem, where a statement is categorized as news or not. We used the *multi-class* classifiers because they give us a more balanced distribution when compared to merging all non-news statements into a single category.

Finally, we opt for *Random Forests* (RF) [Bre01] as our supervised machine learning model. The detailed explanation of RF is shown in Chapter 2. We experimented with other models, but the differences in performance are marginal, and RF have superior learning time. We train the models on the full feature set in Table 5.2.

5.4 Citation Discovery

For the citation discovery task, we follow the *citation policy*[10] guidelines in Wikipedia and single out three key properties on what makes a *good citation.*

1. the statement should be *entailed* by the cited news article
2. the statement should be *central* in the cited news article
3. the cited news article should be from an *authoritative* source

We approach the citation discovery for news statements as follows. We use statement s as a query (see Section 5.4.1) to retrieve the top–k news articles from $\mathcal{N}$ as citation candidates for s. We then classify the candidate citations as either '*correct*' or '*incorrect*', depending on whether they meet the above criteria of a *good citation.*

To do so, we compute features for each pair $\langle s, n_i \rangle$, w.r.t the individual sentences of a news article n_i. The feature vectors become the following: $\langle s, [\sigma_i^1, \sigma_i^2, \ldots, \sigma_i^j] \rangle$, where σ_i^j represents the j-th sentence from n_i.

Since the number of sentences σ_i varies across news articles, we aggregate the individually computed features at sentence level into the corresponding *min, max*, *average*, *weighted average*, and *exponential decay function* scores as shown below.

$$\langle s, \min_j F(\sigma_i^j), \max_j F(\sigma_i^j), Avg(F(\sigma_i)), \sum_{\sigma_i^j} \frac{1}{j} * F, \sum_{\sigma_i^j} F^{\frac{1}{j}}, \ldots \rangle \tag{5.5}$$

where F is a feature from the complete feature list in Table 5.3, 5.4, and Section 5.4.4.

5.4.1 Query Construction

We use the text from the statement s to query our news collection $\mathcal{N}$. The length of a statement can be from a single sentence to an entire paragraph. For this reason, we need to employ efficient query construction approaches that extract the keywords from s such that we increase the coverage and likelihood of retrieving relevant news article candidates for statement s.

It has been shown that in similar cases where the query corresponds to a sentence or paragraph, query construction (QC) approaches are necessary to increase the accuracy of IR models. Henzinger et al. [HCMB03] propose several QC approaches that weigh query terms based on the *tf–idf* score, and other variations of weighting the individual terms.

We experimented with different QC approaches from [HCMB03] and their impact on finding news articles in $\mathcal{N}^W$. From the many variations proposed in [HCMB03], we

[10] https://en.wikipedia.org/wiki/Wikipedia:Citing_sources

find one approach *QCA1Base*, which had the best performance in terms of improving the coverage of relevant articles in top–k. In *QCA1Base*, the terms extracted from the statements are weighted based on *tf–idf*. The score of tf simply represents the term frequency of a word in s, whereas *idf* is computed with respect to other terms in our statement collection.

In principle, one should consider all retrieved articles from the result set. However, this is not only computationally expensive for our subsequent learning step but also unbalances our training set. The reason for this is that, news statements on average do not have more than one citation. Hence, if we go to high retrieval depths in order to have perfect recall, we have the vast majority of documents being irrelevant for s.

To determine a reasonable retrieval depth, we experimented with 1000 randomly chosen statement queries with QC and determined the hit-rate at retrieval depth k , i.e. whether the cited article is retrieved in the top—k articles.

Figure 5.5 shows the hit-rate in top–1000 with top 50 ranked query terms and with *divergence from randomness* (see Chapter 2) query similarity measure [AVR] for our *random sample* of 1000 news statements.

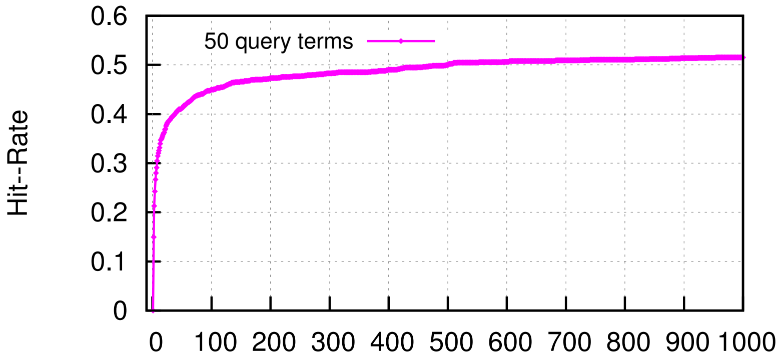


Figure 5.5. Hit-rate of articles in $\mathcal{N}^W$ up to rank 1000 (x–axis) for 1000 news statements, respectively *QCA1Base* queries.

We focus on the top–100 retrieved news articles as potential citations for s, as the achieved hit-rate beyond the top–100 shows only minor improvement. In Figure 5.5, we also note that the hit-rate does not go beyond 50%. We found that most of the news articles that are not retrieved are either missing or non-English articles in $\mathcal{N}^W$.

5.4.2 Textual Entailment Features

As the citation is supposed to give strong evidence for the statement's content, in the ideal case the cited news article should fully *entail* the statement, i.e. the statement should be derivable from the news article. The recognition of textual entailment has been the study of extensive research in the last 10 years; cf [DRSZ13] for an overview.

	entailment
$\mathbf{J}^n(s, \sigma_i^j)$	n–gram overlap between s and σ_i^j and similarity headline of n_i and s
$\mathbf{J}^P(s, \sigma_i^j)$	NNP phrase overlap between s and σ_i^j
$\theta^1(s, \cdot)$	s unigram LM from news article n_i and n–gram LM from news articles in N_t
$K(s, \sigma_i^j)$	tree kernel similarity between s and σ_i^j
$LDA(s, N_t)$	term overlap between s and topic terms from news in N_t
$freq(e)$	occurrence frequency of e in the title and body of n_i
baseline features	retrieval score from the IR model for n_i and its rank

Table 5.3. Entailment feature set for the citation discovery task.

A full treatment of entailment needs extensive world knowledge and inference rules; we here restrict ourselves to much simpler lexical and syntactic similarity methods used in baseline entailment systems and leave the extensions to future work.[11]

IR Baseline Features. We use the retrieval model as a pre-filter to find candidate news articles as citations for s. The retrieval model also provides us with two possible features for the learning model: firstly, a *matching score* of n_i for query s, where the score corresponds to the *divergence from randomness* query similarity measure [AVR]. Secondly, the retrieval rank of n_i. We use the IR model as our *baseline* and hence refer to them as *baselines features*.

Tree Kernel Similarity. Lexical similarity measures in many cases fail to capture the joint *semantic* and *syntactic* similarity. For this purpose, we consider the *tree kernel* similarity measure proposed in [Kat08]. We first compute the *dependency parse trees* of s and σ_i^j using the Stanford tagger [TM00], and then compute the tree kernel, $K(s, \sigma_i^j)$. Tree kernel similarity through the dependency parse tree measures the maximum matching subtrees between s and σ_i^j, where the matching subtrees have the same syntactic and semantic meaning. We refer the reader to [Kat08] for details.

LM & Topic Model Scoring. From an article n_i we compute a *unigram LM* and compute $\theta(s, n_i)$ as the likelihood of s being generated from the computed LM. In addition, we compute *n–gram* LM (with n up to 3) from articles in N_t, and compute the score $\theta^n(s, N_t)$ accordingly.

Similarly, we compute *LDA topic models* [BNJ03] for entity types, specifically from articles in N_t. This follows the intuition that content usually is clustered around specific topics, i.e. for type `Politician` most discussions are centered around politics,

[11] Off-the-shelf entailment systems exist but are too slow to use at scale.

career, etc. The topic score is the Jaccard similarity between n_i and the topic terms.

5.4.3 Centrality Features

	centrality
$\mathbf{J}(s, \sigma_i^c)$	Jaccard similarity between s and central sentence σ_i^c
$\mathbf{J}^P(s, \sigma_i^c)$	NNP phrase overlap between s and σ_i^c
$\mathbf{J}^n(s, \sigma_i^c)$	n–gram overlap between s and σ_i^c
$K(s, \sigma_i^c)$	tree kernel similarity between s and σ_i^c
$\phi(e, n_i)$	the relative entity frequency of e in n_i
$\phi(\gamma(s), n_i)$	relative entity frequency of $e \in \gamma(s)$ in n_i

Table 5.4. Sentence centrality feature set for the citation discovery task.

Similarity to most central news sentence. As described above we compute similarity features between s and sentences in n_i. However, some sentences in n_i are more central than others. Hence, the computed features between the pairs $\langle s, [\sigma_i^1, \sigma_i^2, \dots, \sigma_i^j]\rangle$, do not have uniform weight. Therefore, we find the most *central sentence* σ_i^c in n_i and distinguish the computed entailment/similarity features between s and σ_i^c.

We compute centrality of a sentence in n_i through the TextRank approach introduced in [MT04]. We first construct a graph $G = (V, E)$ from n_i, where V corresponds to the sentences of n_i, with edges in E weighted with the Jaccard similarity between any two sentences, in this case $\sigma_i^j \in V$. Computation of centrality for any vertex σ_i^j is similar to that of PageRank, with slight changes accounting for the weighted edges between vertices.

$$\Gamma(\sigma_i) = (1 - d) + d * \sum_{\sigma_j \in In(\sigma_i)} \frac{\mathbf{J}(\sigma_i, \sigma_j)}{\sum\limits_{\sigma_k \in Out(\sigma_j)} \mathbf{J}(\sigma_j, \sigma_k)} \Gamma(\sigma_j) \tag{5.6}$$

where d is the damping factor ($d = 0.85$), a common value in PageRank computation. The computation converges if the difference in the score of $\Gamma(\sigma_j)$ in two consecutive iterations is small.

Relative Entity Frequency. The importance of e in n_i is crucial when finding citations for s. This importance is partially mirrored simply in how often e is mentioned in n_i. However, another genre-typical property of news is its inverted pyramid structure, i.e. the most important information is mentioned at the beginning of the article. We therefore measure relative entity frequency of e in n_i based on an approach described in Chapter 7. It attributes higher weight to entities appearing in the top paragraphs of n_i, where the weight follows an exponential decay function.

$$\phi(e,n) = \frac{|\rho(e,n)|}{|\rho(n)|} \sum_{\rho \in \rho(n)} \left(\frac{tf(e,\rho)}{\sum\limits_{e} tf(e',\rho)} \right)^{\frac{1}{\rho}} \tag{5.7}$$

where ρ represents a news paragraph from n and $\rho(n)$ indicates the set of all paragraphs. $tf(e,\rho)$ indicates the frequency of e in ρ. With $|\rho(e,n)|$ and $|\rho(n)|$ we indicate the number of paragraphs in which entity e occurs and the total number of paragraphs.

Additionally we consider the relative entity frequency for entities in $e \in \gamma(s)$ and measure the *minimum, maximum* and *average* relative entity frequency scores.

5.4.4 News-Domain Authority Features

Wikipedia's editing policy distinguishes clearly between more and less-established news outlets and prefers the former. We therefore compute the authority of news domains w.r.t entity types and sections.

We will denote the *domain* of the news article referred from s as $D[s]$, and with D any arbitrary domain.

Type-Domain Authority. Authority of news domains is non-uniformly distributed across types. For types such as `Politician` the authority of domains like *BBC* is higher than for types such as `Athletes`, where a domain specialized in sports news is more likely to be authoritative. We capture the *type-domain authority* as follows:

$$p(D|t) = \frac{\sum_{e \in \mathbf{W} \wedge t \in T(e)} \sum_{s \in S(e)} 1_{D=D[s]}}{\sum_{e \in \mathbf{W} \wedge t \in T(e)} \sum_{s \in S(e)} D[s]}$$

Section-Domain Authority. We measure the authority of domains associated to certain entity sections. The density of news references across sections varies heavily. Therefore, it is natural to consider the authority of news domains for a given section.

$$p(D|\psi) = \frac{\sum_{e \in \mathbf{W}} \sum_{s \in S(e,\psi)} 1_{D=D[s]}}{\sum_{e \in \mathbf{W}} \sum_{s \in S(e,\psi)} D[s]}$$

Note that these features compute news outlet authority with regard to current Wikipedia usage, which we seek to re-create. An alternative we intend to look at in future work is to measure authoritativeness via Wikipedia-external measures of news outlets, such as page visits or interlinkage.

5.5 Statement Categorization Evaluation

Here we describe the evaluation of our approach for the statement categorization task. Since we consider the type taxonomy from YAGO, we have a hierarchy of models. Each statement belongs to an entity, which in turn is a child to a type (node) in the hierarchy. Consequently, we construct each model from training instances (statements) that are its children. We focus on two aspects (i) performance of models at varying depths, and (ii) performance of various feature classes.

5.5.1 Experimental Setup

Setup. We consider 672 entity types from our YAGO taxonomy, for which we learn individual SC models. We consider types that have more than 1000 entity instances. The level of granularity in the YAGO taxonomy has a maximum depth of 20, while the root type is `owl:Thing` containing all possible entities.

Train/Test. We learn the SC models using up to 90% of the entity instances of a type t as training set, and the remainder of 10% for evaluation. We use *stratified sampling* to pick entities of type t and its subtypes for the train and test set. We train and test SC models over 6 million statements coming from 1.3 million entities.

Metrics. We evaluate the performance of SC with *precision* P, *recall* R and $F1$. A statement is considered to be classified correctly, if the predicted citation category matches to the citation category in our ground-truth.

5.5.2 Results and Discussion

The following discussion focuses on the results for the statement categorization task for the `news` citation category. We report the classification results for our function SC only for the first three type levels in the YAGO taxonomy, specifically the immediate child `Legal Actor Geo` of `owl:Thing`.[12]

Table 5.5 shows the results for SC models evaluated over 61k entities and trained with up to 550k entities, depending on the training sample size, which we vary in between the ranges $\tau \in [1\%, 90\%]$. The results for the type `yagoLegalActorGeo` represent more than 47% of the total set of entities in our evaluation dataset.

The overall performance of SC for all types for $\tau = 90\%$ measured through *micro-average* precision is 0.57. Since a statement belongs to multiple types $T(s)$, we decide the category of s based on majority as categorized from the individual SC models.

[12] For readability we remove the *wordnet* prefix from the types and their numerical ID values.

yagoLegalActorGeo										
Parent Type	Child Type	$1 \leq \tau \leq 10$			$10 < \tau \leq 50$			$50 < \tau \leq 90$		
		P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
owl:Thing	Legal Actor Geo	0.48	0.36	0.41	0.51	0.43	0.47	0.53	0.47	0.50
Legal Actor Geo	Legal Actor	0.51	0.34	0.41	0.54	0.41	0.47	**0.56**	**0.45**	**0.50**
	location	0.30	0.29	0.29	0.34	0.40	0.37	0.36	0.45	0.40
location	region	0.30	0.28	0.29	0.35	0.40	0.37	0.37	0.44	0.40
	point	0.30	0.1	0.14	0.38	0.22	0.28	0.39	0.26	0.32
Legal Actor	person	0.53	0.36	0.43	0.56	0.43	0.49	**0.58**	**0.46**	**0.51**
person	preserver	0.63	0.31	0.42	0.67	0.46	0.54	**0.67**	**0.49**	**0.57**
	authority	0.53	0.20	0.29	0.62	0.24	0.35	0.65	0.33	0.44
	contestant	0.59	0.43	0.50	0.62	0.52	0.57	0.64	0.56	0.60
	leader	0.53	0.26	0.34	0.59	0.34	0.43	0.61	0.37	0.46
	wc Living people	0.55	0.37	0.44	0.58	0.44	0.50	0.59	0.47	0.52

Table 5.5. Results for the *statement classification* for entities of type yagoLegalActorGeo. Results are for the different sample ranges τ and shown different levels of entity types in the YAGO type hierarchy.

Level of Type Granularity

As expected, we observe that model performance depends on the type level (cf. Table 5.5). A unified model from heterogeneous training instances performs poorly: the *SC* model for the main type Legal Actor Geo achieves a precision P=0.527 with high variance across its subtypes. Comparing the types at depth level 3 (last group of rows), the difference in terms of precision can go as high as 15% between Legal Actor Geo and the best performing subtype preserver.

At higher depths, the performance often improves significantly as the instances belonging to a given type become more homogeneous. For example, the fine grained type wcat Italian footballers has a precision of P=0.87 and recall of R=0.58, which constitutes a 50% precision and a 26% recall improvement over its parent type Person. However, the performance improvement is not monotonically increasing. In some fine-grained types, there is in fact a performance reduction which can be attributed to over-fitting. This suggests that there is indeed a sweet spot in terms of choice of the best performing model for an instance. We observed that the instances that are children of person showed best performances between levels 5 and 8.

Our models perform poorly for types such as location since location pages have a lower news density. We again observe that news articles are usually centered around people and its instances benefit the most from our approach. We also observe that the performance of our approach is sensitive to the type hierarchy. The choice of YAGO as a taxonomy is due its fine-grained types. However, there exist many long-tail entities that are direct descendants from the higher levels and fail to leverage the homogeneity of fine-grained types. We also perform poorly on such instances.

In the YAGO taxonomy, the entities are distributed normally with a mean at depth level 8, which contains around 36% of entities. The long tail with types lower than depth level 8 accounts for 28% of entities in the YAGO taxonomy.

We focussed on the category news in our discussion and in Table 5.5. Performance of *SC* models for the categories $c = \{$web, book, journal$\}$ and type person is P=0.62 and R=0.59, P=0.29 and R=0.69, and P=0.25 and R=0.26, respectively. The relatively high score for the web category can be attributed to the high density of statements of category web, accounting for more than 54% of the total statements. Hence, by always choosing web as the category of a statement we get an average precision of 0.54.

Convergence and Feature Ablation

Convergence. We measure the amount of training data required for the models to converge to optimal performance. Figure 5.6 shows the learning curve for some of the types reported in Table 5.5. We see that *SC* models converge and achieve optimal performance early on with a sample around 7% to 10%.

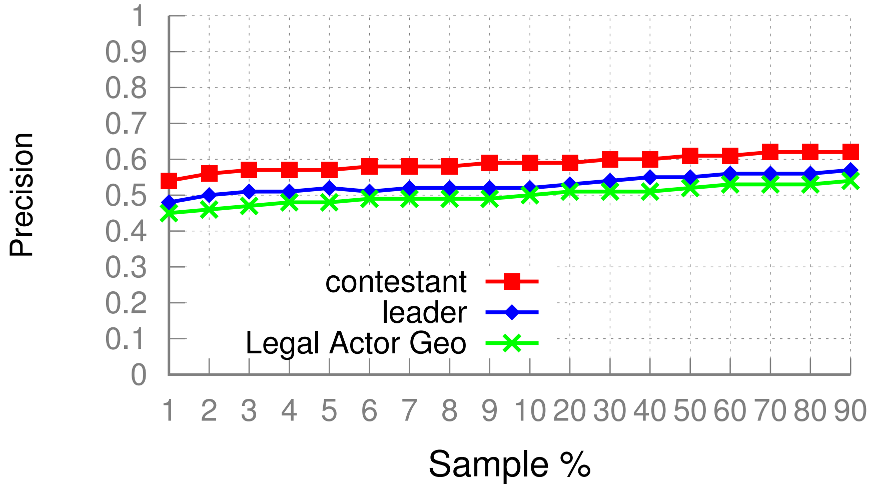


Figure 5.6. Learning curve for *SC* measured for different sample sizes.

Ablation. We apply a feature ablation test for the different different features groups from Section 5.4. Figure 5.7 shows the results for the feature groups *language style*, and *entity structure*. The highest gain is achieved with the feature group *entity structure*, which reveals the challenging nature of the task where language style features cannot be applied alone.

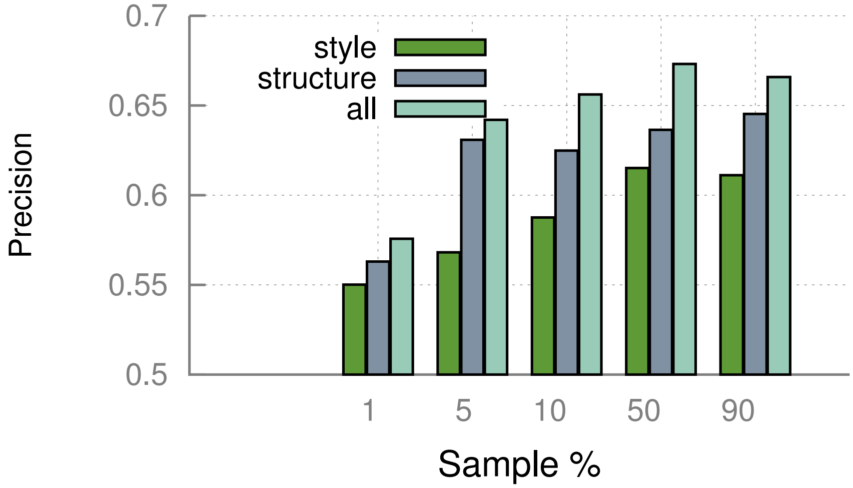


Figure 5.7. Feature ablation for features in *SC* for type `preserver`

5.6 Citation Discovery Evaluation

In this section, we evaluate the *citation discovery* task for *news statements*. We perform an extensive evaluation for approximately 22k news statements and discover citations from a real-word news collection with more than 20 million articles for a timespan of two years.

5.6.1 Statement and News Collection

To establish a realistic evaluation of the citation discovery step, we consider only a subset of our initial Wikipedia news collection $\mathcal{N}^W$ between the year range of 2013 and 2015. The resulting subset contains 22k news statements with 27k news article citations in $\mathcal{N}^W$[13]. We denote this temporal slice of news articles in $\mathcal{N}^W$ by $\mathcal{N}^W_{13-15}$.

The reason for considering only the subset $\mathcal{N}^W_{13-15}$ is because we are in hold of a large news collection, with up to 40k daily news articles, and with more than 20 million news articles between the years 2013 and 2015. The news collection we construct from

[13] A statement can have more than one citation.

the GDelt project[14] and denote it with $\mathcal{N}^G$. The merged news collection of $\mathcal{N}^W_{13-15}$ and $\mathcal{N}^G$ we denote with $\mathcal{N}$.

In $\mathcal{N}$, the collection $\mathcal{N}^W_{13-15}$ accounts for less than 1% of the total news articles, hence, making the task of citation discovery difficult. This represents a realistic scenario, where we are given a large collection of news articles and from the many irrelevant articles we are to find an appropriate citation for a news statement s.

5.6.2 Evaluation Strategies

Here we describe the two evaluation strategies we employ for evaluating the performance of our models. In this task, we could simply count an news article as relevant for a news statement s if it exists in our ground-truth, that is $n \in N_s$. However, given that we are in hold of a high coverage real-world news collection, there are news articles that still may be relevant for s, however, not exist in the ground-truth we extract from Wikipedia. Therefore, we outline two evaluation strategies below.

Evaluation Strategy `E1`: In this scenario, we, for each news statement s, only consider the pairs $\langle s, n \rangle$, where $n \in N_s$ as correct and all other possible citations as incorrect. This allows for fully automatic evaluation but is only a *lower bound* for FC, as there can be additional articles that are relevant for s but do not exist in N_s. We therefore also consider a variant `E1+FP`, where we consider $n' \notin N_s$ as additional correct citations if the similarity (based on the *Jaccard* similarity) to one of the articles in N_s is above 0.8.

Evaluation Strategy `E2`: `E2` assesses the true performance of FC. In this case, apart from already existing citations for s from N_s, we assess through *crowd-sourcing* the appropriateness as citations of articles $n \in \mathcal{N} \wedge n \notin N_s$.

We set up the crowd-sourcing experiment for `E2` as follows. For a statement s and an article $n_i \notin N_s$ marked as correct by FC, we ask the crowd to compare n_i with the ground-truth article $n \in N_s$ and answer the question *'Which of the two shown news articles is an appropriate citation for the statement?'*. The workers are shown s as well as n_i and the ground truth article in random order without an indication which one is the ground truth. We provide the following response options: (i) *first*, (ii) *second*, (iii) *both*, (iv) *none*, and (v) *insufficient info*. We deployed the experiment in CrowdFlower[15] and chose only high quality workers to ensure the reliability of our experiments[16]. Furthermore, we removed workers who did not spend the minimum amount of two minutes to assess the appropriateness of a citation[17].

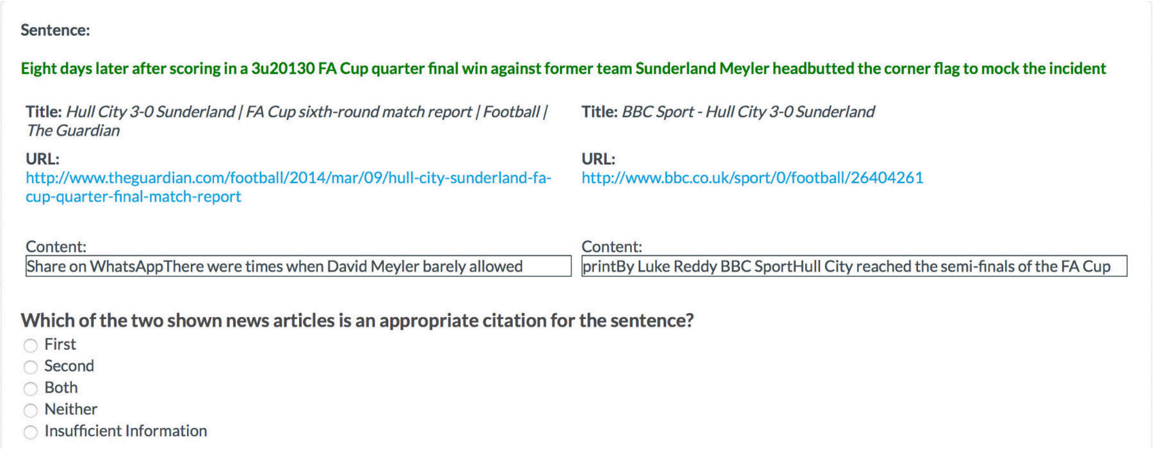
[14] http://gdeltproject.org/

[15] https://www.crowdflower.com

[16] We select workers with the highest quality as provided by the CrowdFlower platform.

[17] The amount of two minutes was decided based on the number of citations the workers had to

We collect three judgments per question. We count citations as correct which are ground-truth articles or articles which the majority of workers judge as appropriate citations. Figure 5.8 shows the visual interface which we display to crowd-workers to assess the relevance of news citation for statement s.



Sentence:

Eight days later after scoring in a 3u20130 FA Cup quarter final win against former team Sunderland Meyler headbutted the corner flag to mock the incident

Title: *Hull City 3-0 Sunderland | FA Cup sixth-round match report | Football | The Guardian*

URL:
http://www.theguardian.com/football/2014/mar/09/hull-city-sunderland-fa-cup-quarter-final-match-report

Content:
Share on WhatsAppThere were times when David Meyler barely allowed

Title: *BBC Sport - Hull City 3-0 Sunderland*

URL:
http://www.bbc.co.uk/sport/0/football/26404261

Content:
printBy Luke Reddy BBC SportHull City reached the semi-finals of the FA Cup

Which of the two shown news articles is an appropriate citation for the sentence?

○ First
○ Second
○ Both
○ Neither
○ Insufficient Information

Figure 5.8. Evaluation interface in CrowdFlower shown to crowd-workers for assessing the relevance of news articles for a news statement s.

5.6.3 Experimental Setup

Retrieval model. We use the retrieval model in [AVR], specifically the implementation provided by Solr[18]. We use the top-100 retrieved news articles for a statement as candidate citations, from which we perform feature extraction and learn our SC models.

Learning Setup. We learn classifiers specific to entity types for a total of 83 types. We limit ourselves to types that have news statements in the date range 2013-2015 and with at least 100 entity instances. From our set of 22k statements, we randomly sample statements from each entity type if they have more than 1000 instances, otherwise we take all statements. Training and testing data consist of the pairs $\langle s, n_i \rangle$, where s is a news statement, and n_i is one of the top–100 citation candidates which we retrieve from $\mathcal{N}$. We split training and testing data per statement s, where each s and all its candidates are included *completely* either in the training or test set.

Learning Approach. We learn the FC models as supervised binary classification models using random forests RF[Bre01]. We predict $\langle s, n \rangle \in$ *'correct'*, *'incorrect'*, i.e.

assess per page (consisting of 5 citations to assess).

[18] http://lucene.apache.org/solr/

if a candidate news article is an appropriate citation for s or not. We optimize for the *'correct'* class. The correct labels in training and automatic evaluation $E1$ are all part of $\mathcal{N}^{W}_{13-15}$, which makes up less than 1% of our news collection N. Therefore, we learn FC as a *cost-sensitive* classifier.

Metrics. We evaluate the performance of FC models through standard evaluation metrics like precision P, recall R, and $F1$ score. A candidate news article n is considered to be a relevant or appropriate citation for the news statement s, if it exists in our ground-truth, that is N_s or is marked by our human evaluators as relevant.

Baselines. We consider two baselines (**B1** and **B2**) for this task. For **B1**, we use the *divergence from randomness* model [AVR] to retrieve news articles from $\mathcal{N}$ for s and simply suggest the top–1 article as citation. In **B2** we learn a supervised model based on the *IR baseline features* (see Table 5.7).

5.6.4 Results and Discussion

Table 5.6 and 5.7 show the results for all evaluation strategies for the citation discovery task. We only show the detailed results for the top–10 best performing entity types out of the 83 types in our evaluation. Please note here that we have only 83 types, the reason for this difference in contrast to the first task of statement categorization is as a results from the constructed news sub-collection $\mathcal{N}^{W}_{13-15}$. That is, the statements that have citations to $\mathcal{N}^{W}_{13-15}$ belong to only 83 types.

The results in each row in Table 5.6 show the best performance we achieve for the individual types, while varying the variables such as the *training sample size* and the number of features, which we select based on Information-Gain feature selection algorithms. We show results with a maximum of 60% training sample size.

We report additionally the overall performance of FC models across all 83 types through *micro-average* in the last row in Table 5.6.

E1: Automated Evaluation

In Table 5.6, in the third column, we show the evaluation results for the strategy `E1`.

Results for `E1` are encouraging given the fact that in top–100 news candidates retrieved from $\mathcal{N}$ only 1% of the news are '*correct*' (on average one relevant citation in $\mathcal{N}^{W}_{13-15}$ per statement). Furthermore, as shown in Figure 5.5 the highest recall we get at top–100 is on average around 45%.

We achieve the best performance in terms of precision for the entity type `football player`, with precision P=0.80 and a recall of R=0.30. For F1 the best performing type in this setup is the entity type `player` with F1=0.57.

Using the evaluation strategy E1+FP, we consider as relevant all *false positive* (FP) articles which are highly similar to the ground-truth articles N_s (above 0.8 similarity). Even though the FP articles do not exist in our ground-truth, the high similarity to the ground-truth article is a strong indicator for them being relevant citations. Using this strategy, the results improve for some of the types with up to 8% in terms of precision. For type entertainer we have an increase of 11%. In absolute numbers, by considering the highly-similar FP articles as relevant we gain an additional 757 news articles out of 12,877, i.e. an additional 6% news citations.

	E1			E1-FP	E2		
type	**P**	**R**	**F1**	**P**	**P**	#f.	%
player	0.67	0.46	0.55	0.71 ▲ (5.63%)	**0.85** ▲ (21.18%)	20	60
entertainer	0.70	0.33	0.45	0.78 ▲ (10.26%)	**0.90** ▲ (22.22%)	40	60
causal agent	0.73	0.28	0.41	0.77 ▲ (5.19%)	**0.88** ▲ (17.05%)	40	60
location	0.55	0.26	0.35	0.62 ▲ (11.29%)	**0.83** ▲ (33.73%)	30	60
artist	0.67	0.21	0.32	0.67	**0.85** ▲ (21.18%)	50	60
football player	0.80	0.30	0.43	0.80	**0.90** ▲ (11.11%)	50	60
wcat Living people	0.67	0.23	0.34	0.70 ▲ (4.29%)	**0.85** ▲ (21.18%)	50	50
creator	0.74	0.25	0.38	0.74	**0.91** ▲ (18.68%)	50	50
organism	0.69	0.30	0.41	0.70 ▲ (1.43%)	**0.83** ▲ (16.87%)	40	60
person	0.64	0.35	0.46	0.66 ▲ (3.03%)	**0.85** ▲ (24.71%)	20	60
micro-average	0.67			0.71 ▲ (5.6%)	**0.86 ▲ (22.00%)**		

Table 5.6. Top–10 best performing entity types for the *FC* task. E1+FP and E2 columns show the improvement for P over E1. Right most column shows the configuration (#f represents the number of top–k most important features, and % is the percentage of training data) for the *FC* models. The last row shows the *micro-average* precision across all *FC* models.

Baselines **B1** and **B2** in Table 5.7 show the difficulty of the citation discovery task. In particular, we show that standard IR models struggle with this task. Choosing only the top–1 article for citation (**B1**) achieves only up to P=0.37. On the other hand, for **B2**, we see that we cannot learn well using only the IR baseline features, and perform even worse than using **B1**.

E2: Automated+Crowdsourced Evaluation

For E2, we report results after re-evaluating performance of *FC* models via gathering judgements for false positive (FP) news articles suggested as citations for s. We

	B1			B2		
type	**P**	**R**	**F1**	**P**	**R**	**F1**
player	0.37	0.36	0.37	0.31	0.28	0.29
entertainer	0.32	0.31	0.31	0.16	0.18	0.17
causal agent	0.26	0.26	0.26	0.17	0.21	0.19
location	0.21	0.19	0.20	0.21	0.23	0.22
artist	0.31	0.31	0.31	0.24	0.27	0.25
football player	0.31	0.31	0.31	0.29	0.38	0.33
wcat Living people	0.27	0.26	0.26	0.21	0.18	0.2
creator	0.34	0.32	0.33	0.25	0.24	0.24
organism	0.29	0.28	0.28	0.22	0.19	0.2
person	0.26	0.24	0.25	0.21	0.23	0.22
micro-average	0.25			0.21		

Table 5.7. The results for the two baselines **B1** and **B2** for the top–10 best performing entity types for the *FC* task. The last row shows the *micro-average* precision across all *FC* models.

evaluate 11,803 false positive news article citation candidates for the top–10 entity types in Table 5.6, from 6.9k news statements. As reported above, crowd-workers could choose between both ground truth and our suggestion being correct, one of them or neither. The inter-rater agreement between workers was 64%. Table 5.8 shows how these false positives were assessed.

both	4,506 (38.2%)
ground truth only	3,768 (31.9%)
our suggestion only	2,287 (19.4%)
neither	1,242 (10.5%)
all	11,803 (100%)

Table 5.8. Relevant citation distribution for E2.

We see that in many cases our suggestion was equal to (38.2%) or even preferred (19.4%) over the ground-truth suggestion. Hence, our method can even improve citation quality in Wikipedia.

In the E2 column in Table 5.6 we show the updated results for *FC* after collecting judgments for false positive news articles. We see that for most of the types we have an average gain of 18% in terms of precision. We achieve the biggest gain of 28% for the entity type location. For the types football player, creator, entertainer, we can suggest news citations with 90-91% precision. Please note that we do not report the recall score for E2, since assessing the appropriateness of every article in $\mathcal{N}$ as a citation for s is not feasible. The recall score is only reported w.r.t the ground-truth articles in $\mathcal{N}^{W}_{13-15}$.

5.7 Pipeline Evaluation

For the evaluation of both tasks in a pipeline scenario, we randomly sample 1000 statements from all categories and ran the process of citation discovery through both steps. Each statement is associated with multiple entity types, as they are extracted from e where $T(e)$ is a set of types. For the *statement categorization* task we perform the evaluation based on our ground-truth; for the *citation discovery* we evaluate the suggested citations as in evaluation strategy `E2`. Note, that here in the evaluation pair we have a news article (that we suggest) and a resource that can be of any type including *book, web, journal*.

Statement Categorization. We set up *statement categorization* as a *majority voting* categorization. For each statement and the type specific classifiers SC we predict the category and pick the category that has the majority of votes. In contrast to the *statement categorization* in Section 5.3, where the original task aimed at showing for which types this task can be performed accurately, we now aim to set up citation discovery in an automated manner.

Based on the ground-truth, 340 out of the 1000 statements were *news statements*. We categorize 368 as news statements, out of which 263 are correct, i.e. P=0.72 and R=0.77. It is interesting to see that we can leverage additional information through *majority voting*, where for the same statement and its associated types we can predict with high accuracy the citation category label of s.

Citation Discovery. For the *citation discovery* task we ran it based on the generic FC model trained on statements belonging to all types, namely `owl:Thing`. We could use the type specific FC, with additional costs for computing type specific features.

In the second task, from the 368 statements classified as news statements, we ran the citation discovery model FC. We are able to suggest 549 news citations for 78 statements. Based on crowd-sourcing evaluation, we suggest 346 relevant citations, i.e. a precision of P=0.63, out of which 200 citations are citations that were preferred over existing ones in the ground-truth. For 146 cases the citations we suggest are considered to be equally appropriate as the existing ones in the ground-truth, for 116 citations the ground-truth ones were preferred over the ones we suggested. Note that our FC models suggest citations for s only in case they fulfill the criteria in Section 5.4, thus, enforcing high accuracy.

5.8 Conclusion

In this chapter, we define and attempt to solve the automatic news citation discovery problem for Wikipedia. We define two tasks – *sentence categorization* and the *citation discovery* – towards finding the correct news citation for a given Wikipedia statement. For the sentence categorization task, we learn a multi-class classifier to predict if a statement requires a news statement. For the news citation discovery problem, we first find the likely candidates by a retrieval model over a real-world news collection followed by a binary classification for the top-ranked candidates.

We find that statement categorization is a hard problem due to lack of context for the NLP-based features to perform well. However, the Wikipedia page and its type structure provide important cues towards accurate classification. On the other hand, we perform well on the citation discovery task with 67% precision (for top-categories) using the automated evaluation, which further improves to over 80% when crowd-sourced. This shows that we not only identify the correct ground truth articles present in Wikipedia, but in some cases our suggestions are a better fit compared to the sources in Wikipedia.

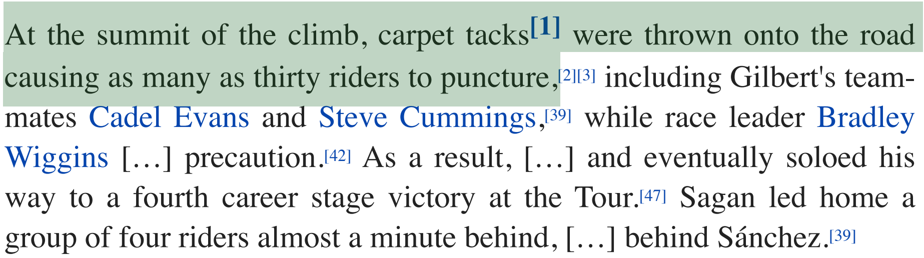
Finally, the contributions in this chapter, are highly important in upholding the core principles of Wikipedia editing policies. Through our proposed approach, we are able to provide citations to statements and as shown in our experimental evaluation, even improve the existing citations further with more authoritative and up-to-date information.

6

Fine Grained Citation Span for References in Wikipedia

Citations uphold the c Citations uphold the crucial policy of *verifiability* in Wikipedia. This policy requires Wikipedia contributors to support their additions with citations from authoritative external sources (web, news, journal etc.). In particular, it states that *"articles should be based on reliable, third-party, published sources with a reputation for fact-checking and accuracy"*[1].

Not only are citations essential in maintaining reliability, neutrality and authoritative assessment of content in such a collaboratively edited platform; but lack of citations are essential signals for core editors for unreliability checks.



At the summit of the climb, carpet tacks[1] were thrown onto the road causing as many as thirty riders to puncture,[2][3] including Gilbert's teammates Cadel Evans and Steve Cummings,[39] while race leader Bradley Wiggins […] precaution.[42] As a result, […] and eventually soloed his way to a fourth career stage victory at the Tour.[47] Sagan led home a group of four riders almost a minute behind, […] behind Sánchez.[39]

Figure 6.1. Sub-sentence level span for citation [1] in a citing paragraph in a Wikipedia article.

However, there are two problems when it comes to citing facts in Wikipedia. First, there is a long tail of Wikipedia pages where citations are missing and hence facts might be unverified. Second, citations might have different span *granularities*, i.e., the text encoding the fact(s), for which a citation is intended, might span less than a sentence (see Figure 6.1) to multiple sentences. We denote the different *pieces of text* which contain a citation marker as *fact statements* or simply *statements*. For

[1] https://en.wikipedia.org/wiki/Wikipedia:Identifying_reliable_sources

example, Table 6.1 shows different *statements* for several citations. The aim of this work is to automatically and accurately determine *citation spans* in order to improve coverage [FMA15, FMNA16] and to assist editors in verifying citation quality at a fine-grained level.

Earlier work on span determination is mostly concerned with scientific texts [O'C82, KTT16], operates at sentence level and exploits explicit authoring cues specific to scientific text. Although Wikipedia has well formed text, it does not follow explicit scientific guidelines for placing citations. Moreover, most statements can only be inferred from the citation text.

In this chapter, we propose a novel approach for determining the citation span for references in Wikipedia. We start from a paragraph that contains either a citation, and operate at the sub-sentence level, that is, we determine the span of citation at the sub-sentence level.

We loosely refer to the sub-sentences as text fragments, and consider a sequence prediction approach using a *linear-chain CRF* [LMP01]. We limit to citations referring to *web* and *news* sources, as they are accessible online and present the most prominent sources in Wikipedia [FAA15]. By using recent work on moving window language models [TW13] and other aspects that take into account the paragraph structure which has a citation, we incrementally attempt to classify continuous text fragments as text that belong to a given citation. We ensure the coherence and accuracy of the selected fragments at all times and for all citation span cases as shown in Table 6.1.

`sub sentence`	Obama was born on August 4, 1961[c1], at Kapi'olani Maternity ··· Honolulu[c2]; he is the first ··· been born in Hawaii.[c3].
`sentence`	He was reelected to the Illinois Senate in 1998, ··· in 2002.[c1]
`multi sentence`	On May 25, 2011, Obama ··· to address ··· UK Parliament in Westminster Hall, London. This was ··· Charles de Gaulle ··· and Pope Benedict XVI.[c1]

Table 6.1. Varying degrees of citation span granularity in Wikipedia text.

By determining the citation span, thus, coupling together with the approach on citation recommendation in the previous chapter, we close a cycle of providing citations for Wikipedia, and consequentially determining at a fine-grained level the validity of such a citation for a given paragraph in Wikipedia.

6.1 Problem Definition and Terminology

In this section, we describe the terminology and define the problem of determining the *citation span* in text in Wikipedia articles.

Terminology. We consider Wikipedia articles $W = \{e_1, \ldots, e_n\}$ from a Wikipedia snapshot. We distinguish *citations* to *external references* in text and denote them with $\langle p_k, c_i \rangle$, where c_i represents a citation which occurs in paragraph p_k with positional index k in an entity $e \in W$. We will refer to p_k as the *citing paragraph*. Furthermore, with *citing sentence* we refer to the sentence in $s \in p_k$, which contains c_i. Note that p_k can have more than one citation as shown in Table 6.1.

Problem Definition. The task of determining the *citation span* for a citation c and a paragraph p, respectively $\langle p, c \rangle$ (or simply p_c), is subject to the citing paragraph and the citation content. In particular, we refer with *citation span* to the *textual fragments* from p which are covered by c. The fragments correspond to the sequence of *sub-sentences* $\mathcal{S}(p) = \langle \delta_1^1, \delta_1^2, \ldots, \delta_1^k, \ldots, \delta_n^m \rangle$. We obtain the sequence of sub-sentences from p by splitting the sentences into sub-sentences or text fragments based on the following punctuation delimiters ($\{,!;:?\}$). These delimitors do not always provide a perfect semantic segmentation of sentences into facts. A more involved approach could be taken akin to work in text summarization, such as Zhou and Hovy [ZH06] or [NPM07] who consider *summary units* for a similar purpose.

Formally we define the *citation span* in Equation 6.4 as the function of finding the subset $\mathcal{S}' \subseteq \mathcal{S}$ where the fragments in $\mathcal{S}'$ are covered by c.

$$\varphi(p, c) \rightarrow \mathcal{S}' \subseteq \mathcal{S}, \quad s.t. \;\; \delta \in \mathcal{S}' \wedge c \vdash \delta \qquad (6.1)$$

where $c \vdash \delta$ states that δ is covered in c.

We note here the subtle difference between a statement as defined in the previous chapter, where we defined it to be as the *inter-citation* text. That is, a statement either started at the beginning of a paragraph or at the end of a sentence which has a citation. We use such definition as one of our baselines for comparison, but as we will see later in the experimental evaluation, this proves to be to coarse grained and accounts for large amount of erroneous spans for a citation.

6.2 Citation Span Approach

We approach the problem of citation span detection in Wikipedia as a *sequence classification* problem. For a citation c and a citing paragraph p, we chunk the paragraph into textual fragments at the *sub-sentence* granularity, shown in Equation 6.4.

Figure 6.2 shows an overview of the sequence classification of textual fragments. We use a *linear chain CRF* [LMP01], where for any fragment δ we predict the label corresponding to a random variable **y** which is either *'covered'* or *'not-covered'*. We opt for CRFs since we can encode global dependencies between the text fragments and the actual citation, thus, ensuring the coherence and accuracy of the predicted labels.

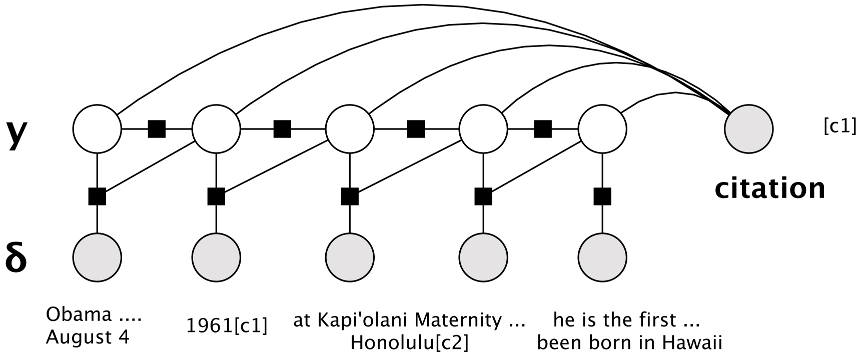


Figure 6.2. Linear chain CRF representing the sequence of text fragments in a paragraph. In the factors we encode the fitness to the given citation.

In the following we describe the features we compute for the factors $\Psi(y_i, y_{i-1}, \delta_i)$ for a sequence δ_i w.r.t the citation c. We determine the fitness of δ_i holding true or being covered by c. We denote with f_k the features for the factors $\Psi_i(y_i, y_{i-1}, \delta_i)$ for sequence δ_i for the linear-chain CRF in Figure 6.2.

6.2.1 Structural Features

An important aspect to consider for the citation the span is the structure of the citing paragraph, and correspondingly its sentences. For a textual fragment δ, we extract the following structural features shown in Table 6.2.

factor	description
$f_i^{c'}$	presence of other citations in δ_i where $c' \neq c$
$f^{\#s}$	the number of sentences in p
$f_i^{\lvert\delta_i\rvert}$	the length in terms of characters of the sub-sequence
f_i^s	check if δ_i is in the same sentence as the citation c
$f_i^{s \neq s'}$	check if δ_i is in the same sentence as δ_{i-1}
f_i^c	the distance of sequence δ_i to the sequence which contains citation c

Table 6.2. Structural features for a sequence δ_i.

From the features in Table 6.2 we highlight f_i^c which specifies the distance of δ to the sequence that cites c. The closer a sequence is to the citation the higher the likelihood of it being covered in c. In Wikipedia, depending on the citation and the paragraph length, the validity of a citation is densely concentrated to its nearby sequences (preceding and succeeding).

Furthermore, the features $f^{\#s}$ and f_i^s, respectively the number of sentences together with the feature considering if δ is in the same sentence as the sequence holding c are strong indicators of predicting accurately the label of δ. That is, it is more likely for a sequence δ to be covered by the citation if it appears in the same sentence or sentences nearby to the citation marker.

However, as shown in Table 6.1 there are three main citation span groups, and as such relying only on the structure of the citing paragraph does not yield optimal results. Hence, in the next group we consider features that tie the individual sequences in the citing paragraph with citation as shown in Figure 6.2.

6.2.2 Citation Features

A core indicator whether a textual fragment δ is covered in c is based on the lexical similarity between δ and the content in c. We gather such evidence by computing two main similarity measures in this case. We compute the features f_i^{LM} and f_i^J between δ and paragraphs in the citation content c.

In details, f_i^{LM} corresponds to a moving language window proposed in [TW13]. In this case, for each word in either a paragraph in the citation c or the sequence δ, we associate a language model M_{w_i} based on its context $\phi(w_i) = \{w_{i-3}, w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}, w_{i+3}\}$ with a window of +/- 3 words. The parameters for the model M_{w_i} are estimated as in Equation 6.2 for all the words in the context $\phi(w_i)$ and their frequencies denoted with tf. With M_δ and M_p we denote the overall models as estimated in Equation 6.2 for the words in the respective fragments.

$$P(w|M_{w_i}) = \frac{tf_{w,\phi(w_i)}}{\sum_{w' \in \phi(w_i)} tf_{w',\phi(w_i)}} \tag{6.2}$$

Finally, we compute the similarity of each word in $w \in \delta$ against the language model of paragraph $p \in c$ in Equation 6.3, which corresponds to the Kullback-Leibler divergence score.

$$f_i^{LM} = \min_{p \in c} \left[- \sum_{w \in \delta} P(w|M_\delta) \log \frac{P(w|M_\delta)}{P(w|M_p)} \right] \tag{6.3}$$

The intuition behind f_i^{LM} is that for the fragments δ we take into account the word similarity and the similarity in the context they appear in w.r.t a paragraph in a citation. In this way we ensure that the similarity is not by chance but is supported by the context in which the word appears. Finally, another advantage of this model is that we localize the paragraphs in c which provide evidence for δ.

Aa an additional feature we compute f_i^J which corresponds to the maximal *jaccard* similarity between δ_i and paragraphs $p \in c$.

Finally, as we will show in our experimental evaluation in Section 6.3, there is a high correlation between the citation span length and the length of a citation content in terms of sentences. Hence, we add as an additional feature f^c the length in terms of the number of sentences for c.

6.2.3 Discourse Features

An indicator which helps on tying sequences in a sentence is the presence of discourse senses.

The discourse senses and the sense type establish if fragments in a sentence are semantically related. We annotate a sentence with explicit discourse senses based on an approach proposed in [PN09]. It relies on the occurrence of discourse cues. The explicit discourse senses belong to one of the following: *temporal, contingency, expansion, comparison.*

After extracting the discourse sense for a sentence, specifically the discourse cue, based on its position we determine to which fragment it belongs and mark the sequence accordingly with the discourse sense. We denote with f_i^{disc} the discourse feature for the sequence δ_i.

6.2.4 Temporal Features

An important aspect that we consider here is the temporal difference between two consecutive fragments δ_i and δ_{i-1}. If there exists a temporal date expression in δ_i and δ_{i-1} and they point to different time-points, this presents an indicator on the transitioning between the states y_i and y_{i-1}. That is, there is a higher likelihood of changing the state in the sequence $\mathcal{S}$ for the labels y_i and y_{i-1}.

We compute the temporal feature here $f_i^{\lambda(i,i-1)}$ indicating the difference in *days* between any two temporal expression extracted from δ_i and δ_{i-1}. We extract the temporal expression through a set of hand-crafted regular expressions. We use the following expressions: (1) `DD Month YYYY`, (2) `DD MM YYYY`, (3) `MM DD YY(YY)`, (4) `YYYY`, with delimiters (whitespace, '-', '.').

6.3 Experimental Setup

We outline the experimental setup for evaluating the citation span approach and the competitors for this task. The data and the proposed approaches are provided as an appendix of this thesis, and can be accessed at the URL[2].

6.3.1 Dataset

We evaluate the citation span approaches on a random sample of Wikipedia entities (snapshot of 20/11/2016). For the sampling process, we first group entities based on the number of citations (*web* or *news*[3]) and sample from the specific groups. This is due to the inherent differences in citation span for entities with varying number of citations.

For instance, entities with high number of citations tend to have shorter span per citation. Figure 6.3 shows the distribution of entities from the different groups. From each sampled entity we extract the *citing paragraphs* that contain either a *web* or *news* citation. Our sample consists of 500 citing paragraphs from 134 entities.

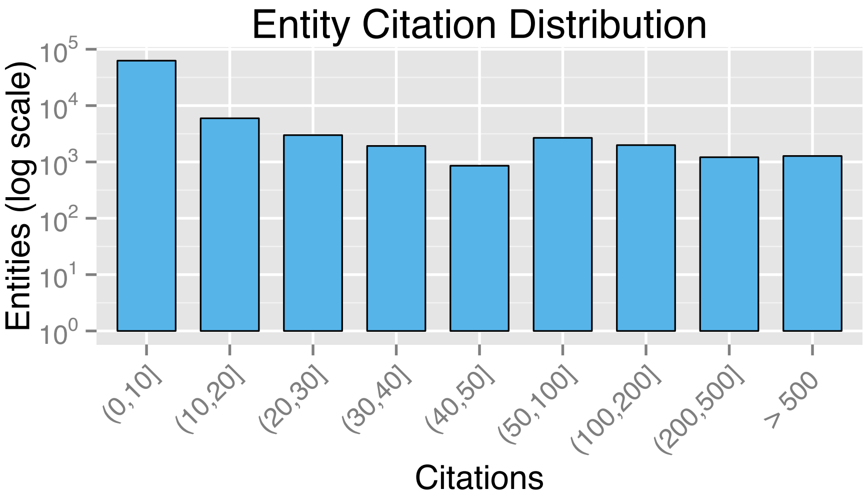


Figure 6.3. Entity distribution based on the number of news citations.

6.3.2 Ground-Truth

Setup. The authors of this work manually determined the citation span of c in paragraph p. The reason for this is due to high evaluation efforts which cannot be enforced in crowdsourcing frameworks, thus raising quality concerns.

We set strict guidelines that help us generate reliable ground-truth annotations. We follow two main guidelines: (i) requirement to read and comprehend the content in c, and (ii) matching of the textual fragments from p as either being supported

[2] http://l3s.de/~fetahu/emnlp17/

[3] Wikipedia has an internal categorization of citation based on the reference they point to.

explicitly or *implied/inferred* in c. In the case where the citation is not appropriate for the paragraph we simply skip such cases[4].

Since the task requires reading and comprehending the entire content in c and p, it takes on average up to 2.4 minutes to perform the evaluation for a single item. It is worth noting that for a textual fragment in p and for citation c we extract *binary* judgements (‘covered‘,‘not-covered’).

Citation Span Stats. Following the definition in Equation 6.4 we determine the citation span at the sub-sentence granularity level. Table 6.3 shows the distribution of citations falling into the specific spans for the citing paragraphs. We note that the majority of citations have a span between half a sentence[5] and up to a sentence, yet, the remainder of more than 20% of citation span across multiple sentences in such paragraphs.

We define the citation span as the ratio of sub-sentences which are covered by a given citation over the total number of sub-sentences in the sentence, consequentially in the citing paragraph. That is, a citation is considered to have a span of one sentence if it covers all its sub-sentences.

$$span(c, p) = \sum_{s \in p} \frac{\#\delta^s \in \mathcal{S}'}{\#\delta^s} \tag{6.4}$$

where δ^s represents a sequence in sentence $s \in p$, which are part of the the ground-truth.

	total	$\leq .5$	$(.5, 1]$	$(1, 2]$	$(2, 5]$	> 5
news	316	0.11	0.64	0.17	0.07	0.02
web	190	0.07	0.64	0.14	0.13	0.03

Table 6.3. Citation span distribution based on the number of sub-sentences in the citing paragraph.

In Figure 6.4 we analyze a possible factor to the variance in the citation span. It is evident that for longer documents the span increases. This is intuitive since such documents carry more information and consequentially their span in the citing paragraphs can be larger. An example is the Wikipedia article `2008 US Open (tennis)` which has a citing paragraph with a citation span of 7 sentences for an article of 30k characters long[6]. We encoded this in the *citation* features f^c.

[4] This is the cases when the language of c is not english.

[5] The sub-sentence span is the ratio of sub-sentences covered over the total number of sub-sentences in a sentence.

[6] http://news.bbc.co.uk/sport1/hi/tennis/7601195.stm

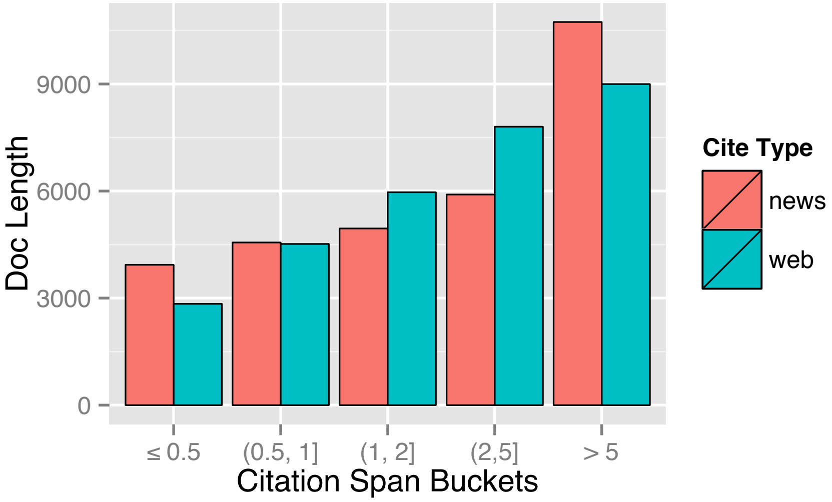


Figure 6.4. Average document length for the different span buckets for citation types *web* and *news*.

Additionally, within the different citation spans we analyze how many of them contain *skips* for two cases: (i) skip a sequence within a sentence, and (ii) skip sentences in p. The results for both cases are presented in Table 6.4. From the results in Table 6.3 and 6.4 we see that simple heuristics on selecting complete sentences or selecting consecutive sequences do not account for the different citation span cases and skips at the sentence and paragraph level. This leads to suboptimal results and introduces erroneous spans.

span	*news*		*web*	
	skip δ	*skip* s	*skip* δ	*skip* s
≤ 0.5	6%	-	-	-
$(0.5, 1]$	-	-	-	1%
$(1, 2]$	-	8%	-	19%
$(2, 5]$	5%	18%	-	21%
> 5	-	20%	-	67%

Table 6.4. The percentage of citations in a span with *sequence skips* and *sentence skips*.

6.3.3 Baselines

We consider the following baselines as competitors for our citation span approach.

Inter-Citation Text – IC. This baseline was employed in the citation recommendation task in Chapter 5. The span in this case consists of sentences which start either at the beginning of the paragraph or at the end of a sentence that contains a citation. The citation span granularity of this baseline is at the sentence level.

Citation-Sentence-Window – CSW. The span consists of sentences in a window of +/- 2 sentences from the citing sentence [O'C82]. The other sentences are included if they contain specific cue words in fixed positions. The cue words are drawn from a fixed vocabulary of 12 words that are specific for scientific literature (e.g. *'above-mentioned', 'these', 'therefore'*, ...).

Citing Sentence – CS. The span consists of only the *citing sentence*. This presents a strong baseline, which based on the distribution of citation span in Table 6.3, around 64% of citations have a span of up to one sentence. However, as we will see later in our evaluation, the amount of erroneous span is large in the other cases where the span is below or more than a sentence.

Markov Random Fields - MRF. MRFs [QR10] model two functions. First, *compatibility*, which measures the similarity of sentences in p, and as such allows to extract non-citing sentences. Second, the *potential*, which measures the similarity between sentences in c with sentences in p. We use the provided implementation by the authors [QR10].

Citation Span Plain – CSPC. A plain classification setup using the features in Section 7.2, where the sequences are classified in isolation. We use Random Forests [Bre01] and evaluate them with 5-fold cross validation.

6.3.4 Citation Span Approach Setup – CSPS

For our approach *CSPS* as mentioned in Section 7.2, we opt for linear-chain CRFs and use the implementation in [Oka07]. We evaluate our models using 5-fold cross validation, and learn the optimal parameters for the CRF model through the L-BFGS approach [LN89].

6.3.5 Evaluation Metrics

We measure the performance of the citation span approaches through the following metrics. We will denote with W' the sampled entities, with $\mathbf{p} = \{p_c, \ldots\}$ (p_c refers to $\langle p, c \rangle$) the set of sampled paragraphs from e, and with $|\mathbf{p}|$ the total items from e.

Mean Average Precision – MAP**.** First, we define *precision* for p_c as the ratio $P(p_c) = |\mathcal{S}' \cap \mathcal{S}^t|/|\mathcal{S}'|$ of fragments present in $\mathcal{S}' \cap \mathcal{S}^t$ over $\mathcal{S}'$. We measure MAP as in Equation 6.5.

$$MAP = \frac{1}{|W'|} \sum_{e \in W'} \frac{\sum_{p_c \in \mathbf{p}} P(p_c)}{|\mathbf{p}|} \tag{6.5}$$

Recall – R**.** We measure the recall for p_c as the ratio $\mathcal{S}' \cap \mathcal{S}^t$ over the all fragments in $\mathcal{S}^t$, $R(p_c) = |\mathcal{S}' \cap \mathcal{S}^t|/|\mathcal{S}^t|$. We average the individual recall scores for $e \in W'$ for the corresponding $\mathbf{p}$.

$$R = \frac{1}{|W'|} \sum_{e \in W'} \frac{\sum_{p_c \in \mathbf{p}} R(p_c)}{|\mathbf{p}|} \tag{6.6}$$

Erroneous Span – Δ**.** We measure the amount of extra *words* or extra *sub-sentences* (denoted with Δ_w and Δ_δ) added by text fragments that are not part of the ground-truth $\mathcal{S}^t$. The ratio is relative to the number of words or sub-sentences in the ground-truth for p_c. We compute Δ_w and Δ_δ in Equation 6.7 and 6.8, respectively.

$$\Delta_w = \frac{1}{|W'|} \sum_{e \in W'} \frac{1}{|\mathbf{p}|} \sum_{p_c \in \mathbf{p}} \frac{\sum_{\delta \in \mathcal{S}' \setminus \mathcal{S}^t} words(\delta)}{\sum_{\delta \in \mathcal{S}^t} words(\delta)} \tag{6.7}$$

$$\Delta_\delta = \frac{1}{|W'|} \sum_{e \in W'} \frac{1}{|\mathbf{p}|} \sum_{p_c \in \mathbf{p}} \frac{|\mathcal{S}' \setminus \mathcal{S}^t|}{|\mathcal{S}^t|} \tag{6.8}$$

6.4 Results and Discussion

In the section, we describe in details the evaluation results. We focus mainly in two main outcomes. The robustness of our approach on determining the citation span at a fine-grained level, and the amount of error introduced while determining the citation span, that is, the amount of *erroneous span*.

6.4.1 Citation Span Robustness

Table 6.5 shows the results for the different approaches on determining the citation span for all span cases shown in Table 6.3.

Accuracy. Not surprisingly, the baseline approaches perform reasonably well. *CS* which selects only the citing sentence achieves a reasonable $MAP = 0.86$ and similar recall. A slightly different baseline *CSW* achieves comparable scores with $MAP = 0.85$. This is due to the inherent span structure in Wikipedia, where a large portion of citations span up to a sentence (see Table 6.3). Therefore, in approximately 64% of the cases the baselines will select the correct span. For the cases where the span is more than a sentence, the drawback of these baselines is in coverage. We show in the next section a detailed decomposition of the results and highlight why even in the simpler cases, a sentence level granularity has its shortcomings due to sequence skips as shown in Table 6.4.

Overall, when comparing *CS* as the best performing baseline against our approach *CSPS*, we achieve an overall score of $MAP = 0.83$ (a slight decrease of 3.6%), whereas in term of F1 score, we have a decrease of 9%. The plain-classification approach *CSPC* achieves similar score with $MAP = 0.86$, whereas in terms of F1 score, we have a decrease of 8%. As described above and as we will see later on in Table 6.6, the overall good performance of the baseline approaches can be attributed to the citation span distribution in our ground-truth.

On the other hand, an interesting observation is that sophisticated approaches, geared towards scientific domains like *MRF* perform poorly. We attribute this to *language style*, i.e., in Wikipedia there are no explicit citation hooks that are present in scientific articles. Comparing to *CSPS*, we outperform *MRF* by a large margin with an increase in *MAP* by 84%.

When comparing the sequence classifier *CSPS* to the plain classifier *CSPC*, we see a marginal difference of 1.3% for *F1*. However, it will become more evident later that classifying jointly the text fragments for the different span buckets, outperforms the plain classification model.

	MAP	R	F1	Δ_w	Δ_δ
MRF	0.45	0.78	0.56	308%	278%
IC	0.72	**0.94**	0.77	113%	115%
CSW	0.85	0.84	**0.82**	38%	31%
CS	**0.86**	0.84	**0.82**	35%	27%
CSPC	**0.86**	0.68	0.76	**26%**	**23%**
CSPS	0.83	0.69	0.75	32%	24%

Table 6.5. Evaluation results for the different citation span approaches.

Erroneous Span. One of the major drawbacks of competing approaches is the granularity at which the span is determined. This leads to erroneous spans. From Table 6.3 we see that approximately in ∼10% of the cases the span is at sub-sentence level, and in 28% the span is more than a sentence.

The best performing baseline CS has an erroneous span of $\Delta_w = 35\%$ and $\Delta_\delta = 27\%$, in terms of extra words and sub-sentences, respectively. That is, nearly half of the determined span is erroneous, or in other words it is not covered in the provided citation. The MRF approach due to its poor MAP score provides the largest erroneous spans with $\Delta_w = 308\%$ and $\Delta_\delta = 278\%$. The amount of erroneous span is unevenly distributed, that is, in cases where the span is not at the sentence level granularity the amount of erroneous span increases. A detailed analysis is provided in the next section.

Contrary to the baselines, for $CSPS$ and similarly for $CSPC$, we achieve the lowest erroneous spans with $\Delta_w = 32\%$ and $\Delta_\delta = 26\%$, and $\Delta_w = 24\%$ and $\Delta_w = 23\%$, respectively.

Compared to the remaining baselines, we achieve an overall relative decrease of 9% for $\Delta_w(CSPS)$, and 34% for $\Delta_w(CSPC)$, when compared to the best performing baseline CS.

From the *skips* in sequences in Table 6.4 and the unsuitability of sentence granularity for citation spans, we analyze the locality of erroneous spans w.r.t to the sequence that contains c, specifically the distribution of erroneous spans *preceding* and *succeeding* it. For the CS baseline, 71% of the total erroneous spans are added by sequences preceding the citing sequence, contrary to 35% which succeed it. In the case of $CSPS$, we have only 9% of erroneous spans (for Δ_δ) preceding the citation.

6.4.2 Citation Spans and Feature Analysis

We now analyze how the approaches perform for the different citation spans in Table 6.3[7]. Additionally, we analyze how our approach $CSPS$ performs when determining the span without access to the content of c.

Citation Spans. Table 6.6 shows the results for the approaches under comparison for all the citation span cases. In the case where the citation spans up to a sentence, that is $(0.5, 1]$, which presents the simplest citation span case, the baselines perform reasonably well. This is due to the heuristics they apply to determine the span, which in all cases includes the *citing sentence*. In terms of $F1$ score, the baseline CS achieves a highly competitive score of $F1 = 0.97$. Our approach $CSPS$ in this case has slight increase of 1% for $F1$ and an increase of 3% for MAP. $CSPC$ achieves a similar performance in this case.

However, for the cases where the span is at the sub-sentence level or across multiple sentences, the performance of baselines drops drastically. In the first bucket (≤ 0.5) which accounts for 9% of ground-truth data, we achieve the highest score with $MAP = 0.87$, though with lower recall than the competitors with $R = 0.56$. The reason for this is that the baselines take complete sentences, thus, having perfect

[7] The models were retrained and tested for the different buckets with 5-fold cross validation.

	≤ 0.5			$(0.5, 1]$			$(1, 2]$			> 2		
	MAP	R	F1	MAP	R	F1	MAP	R	F1	MAP	R	F1
MRF	0.15	0.88	0.27	0.44	0.80	0.61	0.59	0.74	0.57	0.59	0.63	0.55
IC	0.32	**1.00**	0.45	0.77	**0.99**	0.83	0.73	**0.84**	0.74	0.72	**0.81**	**0.73**
CSW	0.38	**1.00**	0.54	0.93	0.98	0.96	0.88	0.54	0.65	0.79	0.34	0.43
CS	0.40	**1.00**	0.56	0.94	0.98	0.97	0.90	0.53	0.65	**0.80**	0.32	0.42
CSPC	0.85	0.53	0.65	**0.96**	0.97	0.97	**0.96**	0.68	0.79	0.71	0.65	0.68
CSPS	**0.87****	0.56	**0.68****	**0.96**	0.98	**0.98**	0.88	0.73	**0.80***	0.74	0.72	0.70
Δ_{F1}			▲21%			0%			▲8%			▼4%

Table 6.6. Evaluation results for the citation span approaches for the different span cases. For the results of *CSPS* we compute the relative increase/decrease of $F1$ score compared to the best result (based on $F1$) from the competitors. We mark in bold the best results for the evaluation metrics, and indicate with ** and * the results which are highly significant ($p < 0.001$) and significant ($p < 0.05$) based on *t-test* statistics when compared to the best performing baselines (CS, IC, CSW, MRF) based on F1 score, respectively.

recall at the cost of accuracy. In terms of $F1$ score we achieve 21% better results than the best performing baseline CS.

For the span of $(1, 2]$ we maintain an overall high accuracy and recall, and have the highest $F1$ score. The improvement is 8% in terms of $F1$ score. Finally, for the last case where the span is more than 2 sentences, we achieve $MAP = 0.74$, a marginal increase of 3%, however with lower recall, which results in an overall decrease of 4% for $F1$. The statistical significance tests are indicated with ** and * in Table 6.6.

Erroneous Span. Figure 6.5 shows the erroneous spans in terms of words for the metric Δ_w for all citation span cases. It is noteworthy that the amount of error can be well beyond 100% due to the ratio of the suggested span and the actual span in our ground-truth, which can be higher.

In the first bucket (span of ≤ 0.5) with granularity less than a sentence, all the competing approaches introduce large erroneous spans. For $CSPS$ we have a $MAP = 0.87$, and consequentially we have the lowest $\Delta_w = 9\%$, while for $CSPC$ we have only $\Delta_w = 11\%$. In contrast, the non-ML competitors introduce a minimum of $\Delta_w(CS) = 182\%$, with MRFs having the highest error. We also perform well in the bucket $(0.5, 1]$. For larger spans, for instance, for $(1, 2]$, we are still slightly better, with roughly 3% less erroneous span when comparing $CSPC$ and CS. However, only in the case of spans with > 2, we perform below the CS baseline. Despite, the smaller

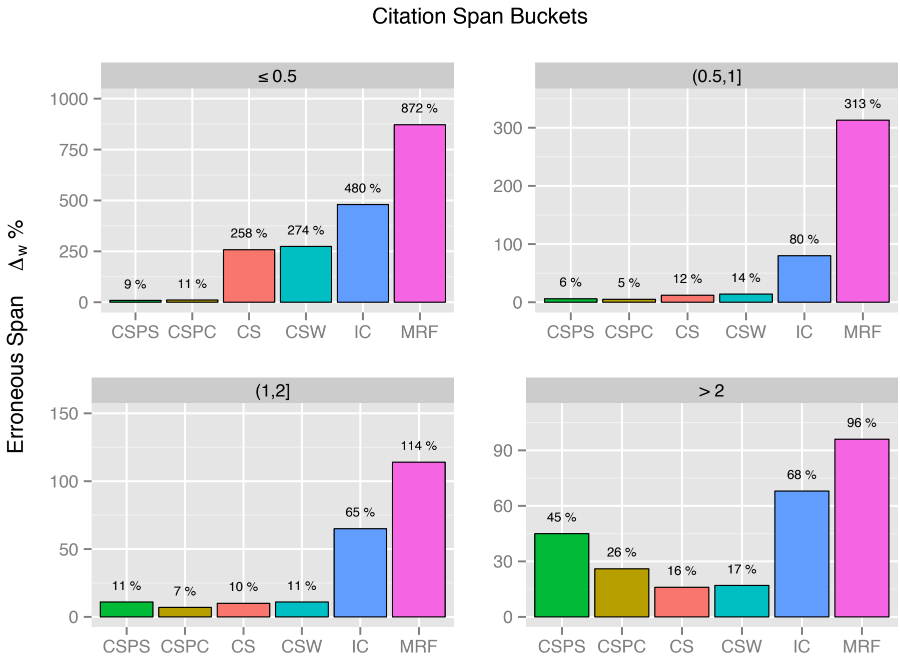


Figure 6.5. Erroneous span for the different citation span buckets. The y-axis presents the Δ_w whereas in the x-axis are shown the different approaches.

erroneous span, the CS baseline never includes more than one sentence, and as such it does not include many erroneous spans for the larger buckets. However, it is by definition unable to recognize any longer spans.

Feature Analysis. It is worthwhile to investigate the performance gains in determining the citation span without analyzing the content of the citation. The reason for this is that there are several citation categories for which access to the source cannot be easily automated. Models which can determine the span accurately without the actual content have the advantage of generalizing to other citation sources (e.g. *books*) for which the evaluation is more challenging.[8]

Here, we disregard the citation features from Section 6.2.2. In terms of MAP, we have a slight decrease with $MAP = 0.82$ when compared to the model with the citation features. For recall we have a drop of 3%, resulting in $R = 0.67$.

This shows that by solely relying on the structure of the citing paragraph and other structural and discourse features we can perform the task with reasonable accuracy.

[8] At worst, one needs to read and comprehend the entire book to determine if a fragment is covered by the citation.

6.5 Citation Span Conclusion

In this work, we tackled the problem of determining the fine-grained citation span of references in Wikipedia. We started from the *citing paragraph* and decomposed it into sequences consisting of sub-sentences. To determine accurately the span we proposed features that leverage the structure of the paragraph, discourse and temporal features, and finally analyzed the similarity between the citing paragraph and the citation content. We limited our approach only to `web` and `news` citations, due to the content of such references being accessible online.

We introduce both a standard classifier as well as a sequence classifier using a linear-chain CRF model. For evaluation we manually annotated a ground-truth dataset of 509 citing paragraphs. We reported standard evaluation metrics and also introduced metrics that measure the amount of erroneous span.

We achieved a $MAP = 0.86$, in the case of the plain classification model $CSPC$, and with a marginal difference for $CSPS$ with $MAP = 0.83$, across all cases with an erroneous span of $\Delta_w = 26\%$ or $\Delta_w = 32\%$, depending on the model. Thus, we provide accurate means on determining the span and at the same time decrease the erroneous span by 34% compared to the best performing baselines. Moreover, we excel at determining citation spans at the sub-sentence level.

In conclusion, this presents an initial attempt on solving the citation span for references in Wikipedia. As future work we foresee a larger ground-truth and more robust approaches which take into account factors such as a reference being irrelevant to a citing paragraph and cases where the evidence for a paragraph is implied rather than explicitly stated in the reference.

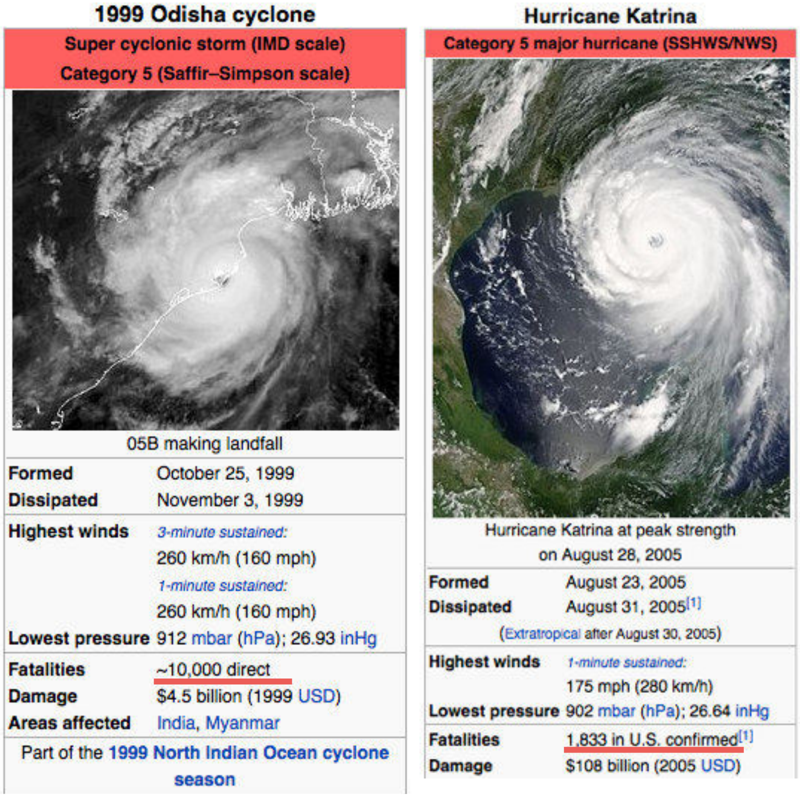
Finally, the contribution in this chapter, when coupled together with the approach on citation recommendation in Chapter 5 closes the cycle on finding citations for Wikipedia entities, and correspondingly determining accurately the span of such citations. This presents a major step on enforcing the Wikipedia editing policies, and enforcing the *verifiability* principle.

7 Automated News Suggestion for Populating Wikipedia Entities

In this chapter, we address the issue of suggesting novel and relevant content coming from news articles to Wikipedia entities. The work in this chapter is partly motivated by our initial study in Chapter 3, where we analyzed the lag between the time facts or general information about an entity is reported in online news media, and the time it is added in Wikipedia. While popular entities and events in Wikipedia are updated instantly [KGC11], updating and maintaining in a timely manner long-tail entities in Wikipedia is subject to extensive research [SB09, BRTN13, BR13, DG14].

In many cases, Wikipedia entity pages are not comprehensive: relevant information can either be *missing* or added with a *delay*. Consider the city of *New Orleans* and the state of *Odisha* which were severely affected by cyclones *Hurricane Katrina* and *Odisha Cyclone*, respectively. While *Katrina* finds extensive mention in the entity page for *New Orleans*, *Odisha Cyclone* which has 5 times more human casualties (cf. Figure 7.1) is not mentioned in the page for *Odisha*. Arguably *Katrina* and *New Orleans* are more popular entities, but *Odisha Cyclone* was also reported extensively in national and international news outlets. This highlights the lack of important facts in trunk and long-tail entity pages, even in the presence of relevant sources. In addition, previous studies have shown that there is an inherent delay or lag when facts are added to entity pages, Chapter 3.

To remedy these problems, it is important to identify information sources that contain novel and salient facts to a given entity page. However, not all information sources are equal. The online presence of major news outlets is an authoritative source due to active editorial control and their articles are also a timely container of facts. In addition, their use is in line with current Wikipedia editing practice, as is shown in [FAA15] that almost 20% of current citations in all entity pages are news articles. We therefore propose *news suggestion* as a novel task that enhances entity pages and reduces delay while keeping its pages authoritative.



1999 Odisha cyclone

Super cyclonic storm (IMD scale)

Category 5 (Saffir–Simpson scale)

05B making landfall

Formed	October 25, 1999
Dissipated	November 3, 1999
Highest winds	3-minute sustained: 260 km/h (160 mph) 1-minute sustained: 260 km/h (160 mph)
Lowest pressure	912 mbar (hPa); 26.93 inHg
Fatalities	~10,000 direct
Damage	$4.5 billion (1999 USD)
Areas affected	India, Myanmar

Part of the 1999 North Indian Ocean cyclone season

Hurricane Katrina

Category 5 major hurricane (SSHWS/NWS)

Hurricane Katrina at peak strength on August 28, 2005

Formed	August 23, 2005
Dissipated	August 31, 2005[1] (Extratropical after August 30, 2005)
Highest winds	1-minute sustained: 175 mph (280 km/h)
Lowest pressure	902 mbar (hPa); 26.64 inHg
Fatalities	1,833 in U.S. confirmed[1]
Damage	$108 billion (2005 USD)

Figure 7.1. Comparing how cyclones are reported in Wikipedia entity pages.

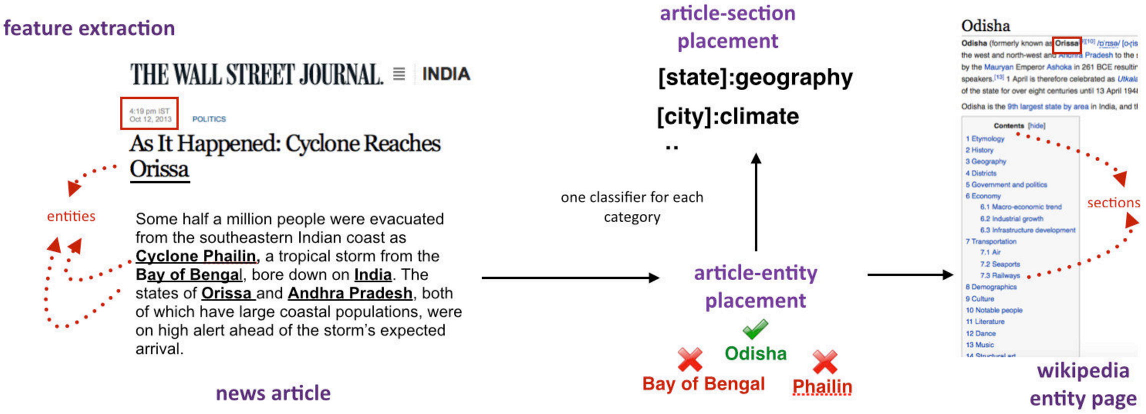
Existing efforts to populate Wikipedia [SB09] start from an entity page and then generate candidate documents about this entity using an external search engine (and then post-process them). However, such an approach lacks in (a) reproducibility since rankings vary with time with obvious bias to recent news (b) maintainability since document acquisition for each entity has to be periodically performed. To this effect, our news suggestion considers a news article as input, and determines if it is valuable for Wikipedia. Specifically, given an input news article n and a state of Wikipedia, the news suggestion problem identifies the entities mentioned in n whose entity pages can improve upon suggesting n. Most of the works on knowledge base acceleration [BRTN13, BR13, DG14], or Wikipedia page generation [SB09] rely on high quality input sources which are then utilized to extract textual facts for Wikipedia page population. In this work, we do not suggest snippets or paraphrases but rather entire articles which have a high potential importance for entity pages. These suggested news articles could be consequently used for extraction, summarization or population either manually or automatically – all of which rely on high quality and relevant input sources.

We identify four properties of good news recommendations: *salience*, *relative authority*, *novelty* and *placement*. First, we need to identify the most salient entities in a news article. This is done to avoid pollution of entity pages with only marginally related news. Second, we need to determine whether the news is important to the entity as only the most relevant news should be added to a precise reference work. To do this, we compute the *relative authority* of all entities in the news article: we call an entity more authoritative than another if it is more popular or noteworthy in the real world. Entities with very high authority have many news items associated with them and only the most relevant of these should be included in Wikipedia whereas for entities of lower authority the threshold for inclusion of a news article will be lower. Third, a good recommendation should be able to identify *novel* news by minimizing redundancy coming from multiple news articles. Finally, addition of facts is facilitated if the recommendations are fine-grained, i.e., recommendations are made on the section level rather than the page level (*placement*).

Approach and Contributions. We propose a two-stage news suggestion approach to entity pages. In the first stage, we determine whether a news article should be suggested for an entity, based on the entity's *salience* in the news article, its *relative authority* and the *novelty* of the article to the entity page. The second stage takes into account the class of the entity for which the news is suggested and constructs *section templates* from entities of the same class. The generation of such templates has the advantage of suggesting and expanding entity pages that do not have a complete section structure in Wikipedia, explicitly addressing long-tail and trunk entities. Afterwards, based on the constructed template our method determines the best fit for the news article with one of the sections.

We evaluate the proposed approach on a news corpus consisting of 351,982 articles crawled from the *news* references in Wikipedia from 73,734 entity pages. Given the Wikipedia snapshot at a given year (in our case [2009-2014]), we suggest news articles that might be cited in the coming years. The existing news references in the entity pages along with their reference date act as our ground-truth to evaluate our approach. In summary, we make the following contributions.

- we propose a two-stage news suggestion approach for Wikipedia entity pages.
- we adopt and address the problem of determining whether a news article should be referenced to an entity considering the entity *salience*, *relative authority* and *novelty* of the article for the entity page.
- we are able to place articles in a specific section of the entity page. Through *section templates*, we address the problems of entities with a limited section structure by class-based generalization i.e. we can expand entity pages with sections that come from entities of a similar class.
- an extensive evaluation on 351,982 news articles and 73,734 entity pages, using their state for the years [2009-2013].



As It Happened: Cyclone Reaches Orissa

Some half a million people were evacuated from the southeastern Indian coast as Cyclone Phailin, a tropical storm from the Bay of Bengal, bore down on India. The states of Orissa and Andhra Pradesh, both of which have large coastal populations, were on high alert ahead of the storm's expected arrival.

Odisha (formerly known as Orissa)[10] /ɒˈrɪsə/ [o-(is

the west and north-west and Andhra Pradesh to the s

by the Mauryan Emperor Ashoka in 261 BCE resultin

speakers.[13] 1 April is therefore celebrated as Utkala

of the state for over eight centuries until 13 April 194

Odisha is the 9th largest state by area in India, and th

Figure 7.2. News suggestion approach overview.

7.1 Problem Definition and Approach Outline

7.1.1 Terminology and Problem Definition

We are interested in named entities e mentioned in news articles. We canonicalize these mentions to entity pages in Wikipedia, a method typically known as *entity linking*. We denote the set of entities extracted and linked from a news article n as $\varphi(n)$. For example, in Figure 7.2, entities are linked to Wikipedia entity pages (e.g. *Odisha* is linked to the Wikipedia entity[1]). For a collection of news articles $\mathbf{N}$, we further denote the resulting set of entities by $\mathbf{E} = \cup_{n \in \mathbf{N}} \{e_i\}$.

Information in Wikipedia entities is organized into sections and evolves with time as more content is added. We refer to the state of Wikipedia at a time t as $\mathcal{W}_t$ and the set of sections for an entity page e as its *entity profile* $S_e(t)$. Unlike news articles, text in Wikipedia could be explicitly linked to entity pages through anchors. The set of entities explicitly referred in text from section $s \in S_e(t)$ is defined as $\gamma(s)$. Furthermore, Wikipedia induces a category structure over its entities, which is exploited by knowledge bases like YAGO (e.g. *Barack_Obama* `isA Person`). Consequently, each entity page belongs to one or more entity categories or classes c. Now we can define our news suggestion problem below:

Definition 7.1 (News Suggestion Problem)**.** *Given a set of news articles* $\mathbf{N} = \{n_1, \ldots, n_k\}$ *and set of Wikipedia entity pages* $\mathbf{E} = \{e_1, \ldots, e_m\}$ *(from* $\mathcal{W}_t$*) we intend to suggest a news article* n *published at time* $t_i > t$ *to entity page* e *and additionally to the most relevant section for the entity page* $s \in S_e(t)$.

[1] http://en.wikipedia.org/wiki/Odisha

7.1.2 Approach Overview

We approach the news suggestion problem by decomposing it into two tasks:

1. *AEP*: *Article–Entity* placement
2. *ASP*: *Article–Section* placement

In this first step, for a given entity-news pair $\langle n, e \rangle$, we determine whether the given news article $n \in \mathbf{N}$ should be suggested (we will refer to this as *'relevant'*) to entity $e \in \mathbf{E}$. To generate such $\langle n, e \rangle$ pairs, we perform the *entity linking* process, $\varphi(n)$, for n.

The *article–entity* placement task (described in detail in Section 7.2.1) for a pair $\langle n, e \rangle$ outputs a binary label (either *'non-relevant'* or *'relevant'*) and is formalized in Equation 7.1.

$$AEP : \langle e, n \rangle \rightarrow \{0, 1\}, \quad \forall e \in \varphi(n) \ \wedge \ n \in \mathbf{N} \tag{7.1}$$

In the second step, we take into account all *'relevant'* pairs $\langle n, e \rangle$ and find the correct *section* for article n in entity e, respectively its profile $S_e(t)$ (see Section 7.2.2). The *article–section* placement task, determines the correct section for the triple $\langle n, e, S_e(t) \rangle$, and is formalized in Equation 7.2.

$$ASP : \langle e, n, S_e(t) \rangle \rightarrow \{s_1, \ldots, s_k\}, \ s \in S_e(t) \tag{7.2}$$

In the subsequent sections we describe in details how we approach the two tasks for suggesting news articles to entity pages.

7.2 News Article Suggestion

In this section, we provide an overview of the *news suggestion* approach to Wikipedia entity pages (see Figure 7.2). The approach is split into two tasks: (i) *article-entity* (*AEP*) and (ii) *article-section* (*ASP*) placement. For a Wikipedia snapshot $\mathcal{W}_t$ and a news corpus $\mathbf{N}$, we first determine which news articles should be suggested to an entity e. We will denote our approach for *AEP* by $\mathcal{F}_e$. Finally, we determine the most appropriate section for the *ASP* task and we denote our approach with $\mathcal{F}_s$.

In the following, we describe the process of learning the functions $\mathcal{F}_e$ and $\mathcal{F}_s$. We introduce features for the learning process, which encode information regarding the entity *salience*, *relative authority* and *novelty* in the case of AEP task. For the *ASP* task, we measure the *overall fit* of an article to the entity sections, with the entity being an input from *AEP* task. Additionally, considering that the entity profiles $S_e(t)$ are incomplete, in the case of a missing section we suggest and expand the entity profiles based on *section templates* generated from entities of the same class c (see Section 7.2.2).

7.2.1 Article–Entity Placement

In this step we learn the function $\mathcal{F}_e$ to correctly determine whether n should be suggested for e, basically a binary classification model (0=*'non-relevant'* and 1=*'relevant'*). Note that we are mainly interested in finding the *relevant* pairs in this task. For every news article, the number of disambiguated entities is around 30 (but n is suggested for only two of them on average). Therefore, the distribution of *'non-relevant'* and *'relevant'* pairs is skewed towards the earlier, and by simply choosing the *'non-relevant'* label we can achieve a high accuracy for $\mathcal{F}_e$. Finding the relevant pairs is therefore a considerable challenge.

An article n is suggested to e by our function $\mathcal{F}_e$ if it fulfills the following properties. The entity e is *salient* in n (a central concept), therefore ensuring that n is about e and that e is important for n. Next, given the fact there might be many articles in which e is *salient*, we also look at the reverse property, namely whether n is important for e. We do this by comparing the *authority* of e (which is a measure of popularity of an entity, such as its frequency of mention in a whole corpus) with the authority of its co-occurring entities in $\varphi(n)$, leading to a feature we call *relative authority*. The intuition is that for an entity that has overall lower authority than its co-occurring entities, a news article is more easily of importance.[2] Finally, if the article we are about to suggest is already covered in the entity profile $S_e(t)$, we do not wish to suggest *redundant* information, hence the *novelty*. Hence, the learning objective of $\mathcal{F}_e$ should fulfill the following properties. Table 7.1 shows a summary of the computed features for $\mathcal{F}_e$.

1. **Salience:** entity e should be a *salient* entity in news article n
2. **Relative Authority:** the set of entities $e' \in \varphi(n)$ with which e co-occurs should have higher *authority* than e, making n important for e
3. **Novelty:** news article n should provide *novel* information for entity e taking into account its profile $S_e(t-1)$

Salience-based features

Baseline Features. A variety of features that measure salience of an entity in text are available from the NLP community. We reimplemented the ones in Dunietz and Gillick [DG14]. This includes a variety of features, e.g. positional features, occurrence frequency and the internal POS structure of the entity and the sentence it occurs in. Table 2 in [DG14] gives details.

[2] This is why people occurring infrequently in the news keep any press cutting mentioning them.

feature	description	
$\Phi(e,n)$	the relative frequency of e in news article n.	*salience*
`Baseline Features`	set of features as proposed by Dunietz and Gillick [DG14]	
$\widehat{\Gamma}(e \mid \varphi(n))$	relative authority as the score of entities that have higher authority than e and that co-occur in n.	*authority*
$P(D)$	measures the news domain authority.	
$\mathcal{N}(n \mid e)$	measures the novelty of a news article n for a given entity e	*novelty*

Table 7.1. *Article–Entity* placement feature summary.

Relative Entity Frequency. Although frequency of mention and positional features play some role in baseline features, their interaction is not modeled by a single feature nor do the positional features encode more than sentence position. We therefore suggest a novel feature called *relative entity frequency*, $\Phi(e,n)$, that has three properties.: (i) It rewards entities for occurring throughout the text instead of only in some parts of the text, measured by the number of paragraphs it occurs in (ii) it rewards entities that occur more frequently in the opening paragraphs of an article as we model $\Phi(e,n)$ as an *exponential decay* function. The decay corresponds to the positional index of the news paragraph. This is inspired by the news-specific discourse structure that tends to give short summaries of the most important facts and entities in the opening paragraphs. (iii) it compares entity frequency to the frequency of its co-occurring mentions as the weight of an entity appearing in a specific paragraph, normalized by the sum of the frequencies of other entities in $\varphi(n)$.

$$\Phi(e,n) = \frac{|p(e,n)|}{|p(n)|} \sum_{p \in p(n)} \left(\frac{tf(e,p)}{\sum\limits_{e'} tf(e',p)} \right)^{p} \tag{7.3}$$

where, p represents a news paragraph from n, and with $p(n)$ we indicate the set of all paragraphs in n. The frequency of e in a paragraph p is denoted by $tf(e,p)$. With $|p(e,n)|$ and $|p(n)|$ we indicate the number of paragraphs in which entity e occurs, and the total number of paragraphs, respectively.

Authority-based features

Relative Authority. In this case, we consider the comparative relevance of the news article to the different entities occurring in it. As an example, let us consider the meeting of the Sudanese bishop *Elias Taban*[3] with *Hillary Clinton*[4]. Both entities are salient for the meeting. However, in Taban's Wikipedia page, this meeting is discussed prominently with a corresponding news reference[5], whereas in Hillary Clinton's Wikipedia page it is not reported at all. We believe this is not just an omission in Clinton's page but mirrors the fact that for the lesser known Taban the meeting is big news whereas for the more famous Clinton these kind of meetings are a regular occurrence, not all of which can be reported in what is supposed to be a selection of the most important events for her. Therefore, if two entities co-occur, the news is more relevant for the entity with the lower a priori authority.

The *a priori authority* of an entity (denoted by $\Gamma(e)$) can be measured in several ways. We opt for two approaches: (i) probability of entity e occurring in the corpus $\mathbf{N}$, and (ii) authority assessed through centrality measures like PageRank [PBMW99]. For the second case we construct the graph $G = (V, E)$ consisting of entities in $\mathbf{E}$ and news articles in $\mathbf{N}$ as *vertices*. The *edges* are established between n and entities in $\varphi(n)$, that is $\langle n \rightarrow \varphi(n) \rangle$, and the out-links from e, that is $\langle e \rightarrow \gamma(s_(t-1)) \rangle$ (arrows present the *edge* direction).

Starting from a priori authority, we proceed to *relative authority* by comparing the a priori authority of co-occurring entities in $\varphi(n)$. We define the *relative authority* of e as the proportion of co-occurring entities $e' \in \varphi(n)$ that have a higher a priori authority than e (see Equation 7.4.

$$\hat{\Gamma}(e|\varphi(n)) = \frac{1}{|\varphi(n)|} \sum_{e' \in \varphi(n)} \mathbb{1}_{\Gamma(e') > \Gamma(e)} \tag{7.4}$$

As we might run the danger of not suggesting any news articles for entities with very high a priori authority, due to the strict inequality constraint, we can relax the constraint such that the authority of co-occurring entities is above a certain threshold.

News Domain Authority. The news domain authority addresses two main aspects. Firstly, if bundled together with the *relative authority* feature, we can ensure that dependent on the entity authority, we suggest news from authoritative sources, hence ensuring the quality of suggested articles. The second aspect is in a news streaming scenario where multiple news domains report the same event — ideally only articles coming from authoritative sources would fulfill the conditions for the news suggestion task.

The *news domain* authority is computed based on the number of news references in Wikipedia coming from a particular *news domain* D. This represents a simple prior that a news article n is from domain D in corpus $\mathbf{N}$. We extract the domains by taking the base URLs from the news article URLs.

[3] http://en.wikipedia.org/wiki/Elias_Taban

[4] http://en.wikipedia.org/wiki/Hillary_Clinton

[5] http://tinyurl.com/mshf7j2

Novelty-based features

An important feature when suggesting an article n to an entity e is the *novelty* of n w.r.t the already existing entity profile $S_e(t-1)$. Studies [BZ05] have shown that on comparable collections to ours (TREC GOV2) the number of duplicates can go up to 17%. This figure is likely higher for major events concerning highly authoritative entities on which all news media will report.

Given an entity e and the already added news references $N_{t-1} = \{n_1, \ldots, n_k\}$ up to year $t-1$, the *novelty* of n_{k+1} at year t is measured by the KL divergence between the language model of n_{k+1} and articles in N_{t-1}. We combine this measure with the *entity* overlap of n_{k+1} and $n' \in N_{t-1}$. The *novelty* value of n_{k+1} is given by the minimal divergence value. Low scores indicate low novelty for the entity profile $S_e(t)$.

$$\mathcal{N}(n|e) = \min_{n' \in N_{t-1}} \{\lambda \cdot D_{KL}\left(\theta(n')||\theta(n)\right) + (1-\lambda) \cdot J\left(\varphi(n'), \varphi(n)\right)\} \tag{7.5}$$

where D_{KL} is the KL divergence of the language models ($\theta(n)$ and $\theta(n')$), whereas λ is the mixing weight ($\lambda = \{0, \ldots, 1\}$) between the language models D_{KL} and the entity overlap in n and n'.

7.2.2 Article–Section Placement

We model the *ASP* placement task as a successor of the *AEP* task. For all the *'relevant'* news entity pairs, the task is to determine the correct entity section. Each section in a Wikipedia entity page represents a different topic. For example, *Barack Obama* has the sections *'Early Life'*, *'Presidency'*, *'Family and Personal Life'* etc. However, many entity pages have an incomplete section structure. Incomplete or missing sections are due to two Wikipedia properties. First, long-tail entities miss information and sections due to their lack of popularity. Second, for all entities whether popular or not, certain sections might occur for the first time due to real world developments. As an example, the entity *Germanwings* did not have an *'Accidents'* section before this year's disaster, which was the first in the history of the airline.

Even if sections are missing for certain entities, similar sections usually occur in other entities of the same class (e.g. other airlines had disasters and therefore their pages have an accidents section). We exploit such homogeneity of section structure and construct templates that we use to expand entity profiles. The learning objective for $\mathcal{F}_s$ takes into account the following properties:

1. **Section-templates:** account for incomplete section structure for an entity profile $S_e(t)$ by constructing section templates $\widehat{S}_c$ from an entity class c

2. **Overall fit:** measures the overall fit of a news article to sections in the section templates $\widehat{S}_c$

feature type	feature	description
Topic	$J(LDA(n), LDA(s_{t-1}))$ $J(LDA(n), N_{t-1})$	Topic similarity between an article n and the section text, and with the already referenced news articles in s.
Syntactic	POS	POS tag overlap (uni/bi/trigrams) between n and s.
Lexical	$J(title(n), s_{t-1})$ $D_{KL}(\theta(p(k)\vert\vert\theta(s_{t-1}))$ $cos(p(n), s_{t-1})$	News title and top–k paragraphs ($k = 1 \dots 5$) similarity with s.
Entity-based	$J(\varphi(n), \gamma(s, t-1))$ $J(\texttt{type}(\varphi(n)), \texttt{type}(\gamma(s_{t-1})))$	Entity and entity type overlap between the news article and entities appearing in a section.
Freq.	$\texttt{\#POS,} \texttt{\vert p(n)\vert,} \vert n\vert, \vert\varphi(n)\vert$ $\texttt{top-}k(e)\texttt{, top-}k(\texttt{type}(e))$	Frequency based features of POS tags, number of paragraphs, entities in a news article

Table 7.2. Feature types used in $\mathcal{F}_s$ for suggesting news articles into the entity sections. We compute the features for all $s \in \widehat{S}_c(t-1)$ as well as s_{t-1}.

Section-Template Generation

Given the fact that *entity profiles* are often incomplete, we construct *section templates* for every *entity class*. We group entities based on their class c and construct *section templates* $\widehat{S}_c$. For different entity classes, e.g. `Person` and `Location`, the section structure and the information represented in those section varies heavily. Therefore, the section templates are with respect to the individual classes in our experimental setup (see Figure 7.3).

$$\widehat{S}_c = \{s_1, \dots, s_k\}, \forall S_e(t) \in \mathbf{E} \wedge e \ \texttt{typeOf} \ c \tag{7.6}$$

Generating *section templates* has two main advantages. Firstly, by considering class-based profiles, we can overcome the problem of incomplete individual entity profiles and thereby are able to suggest news articles to sections that do not yet exist in a specific entity $S_e(t)$. The second advantage is that we are able to canonicalize the sections, i.e. *'Early Life'* and *'Early Life and Childhood'* would be treated similarly.

To generate the section template $\widehat{S}_c$, we extract all sections from entities of a given type c at year t. Next, we cluster the entity sections, based on an extended version of *k–means* clustering [KMN+02], namely *x–means* clustering introduced in Pelleg et al. which estimates the number of clusters efficiently [PM+00]. As a similarity metric we use the cosine similarity computed based on the *tf–idf* models of the sections. Using the *x–means* algorithm we overcome the requirement to provide the number of clusters k beforehand. *x–means* extends the *k–means* algorithm, such that a user only specifies a range $[K_{min}, K_{max}]$ that the number of clusters may reasonably lie in.

News-section fit

The learning objective of $\mathcal{F}_s$ is to determine the overall fit of a news article n to one of the sections in a given section template $\widehat{S}_c$. The template is pre-determined by the class of the entity for which the news is suggested as relevant by $\mathcal{F}_e$. In all cases, we measure how well n fits each of the sections $s \in \widehat{S}_c(t-1)$ as well as the specific entity section $s' \in S_e(t-1)$. The section profiles in $\widehat{S}_c(t-1)$ represent the aggregated entity profiles from all entities of class c at year $t-1$.

To learn $\mathcal{F}_s$ we rely on a variety of features that consider several similarity aspects as shown in Table 7.2. For the sake of simplicity we do not make the distinction in Table 7.2 between the individual entity section and class-based section similarities, $s_e(t-1)$ and $s(t-1)$, respectively. Bear in mind that an entity section s_e might be present at year t but not at year $t-1$ (see for more details the discussion on entity profile expansion in Section 7.4.2).

Topic. We use topic similarities to ensure (i) that the content of n fits topic-wise with a specific section text and (ii) that it has a similar topic to previously referred news articles in that section. In a pre-processing stage we compute the topic models for the news articles, entity sections $S_e(t-1)$ and the aggregated class-based sections in $\widehat{S}_c$. The topic models are computed using LDA [BNJ03]. We only computed a single topic per article/section as we are only interested in topic term overlaps between article and sections. We distinguish two main features: the first feature measures the overlap of topic terms between n and the entity section $s_e(t-1)$ and $s(t-1) \in \widehat{S}_c$, and the second feature measures the overlap of the topic model of n against referred news articles in N_{t-1} at time $t-1$.

Syntactic. These features represent a mechanism for conveying the importance of a specific text snippet, solely based on the frequency of specific POS tags (i.e. `NNP, CD` etc.), as commonly used in text summarization tasks. Following the same intuition as in [SB09], we weigh the importance of articles by the count of specific POS tags. We expect that for different sections, the importance of POS tags will vary. We measure the similarity of POS tags in a news article against the section text. Additionally, we consider *bi-gram* and *tri-gram* POS tag overlap. This exploits similarity in syntactical patterns between the news and section text.

Lexical. As *lexical* features, we measure the similarity of n against the entity section text $s_e(t-1)$ and the aggregate section text $s(t-1)$. Further, we distinguish between the overall similarity of n and that of the different news paragraphs ($p(n)$ which denotes the paragraphs of n up to the 5th paragraph). A higher similarity on the first paragraphs represents a more confident indicator that n should be suggested to a specific section s. We measure the similarity based on two metrics: (i) the KL-divergence between the computed *language models* and (ii) *cosine* similarity of the corresponding paragraph text $p(n)$ and section text.

Entity-based. Another feature set we consider is the overlap of *named entities* and their corresponding *entity classes*. For different entity sections, we expect to find a particular set of entity classes that will correlate with the section, e.g. '*Early Life*' contains mostly entities related to family, school, universities etc.

Frequency. Finally, we gather statistics about the number of entities, paragraphs, news article length, top–k entities and entity classes, and the frequency of different POS tags. Here we try to capture patterns of articles that are usually cited in specific sections.

7.3 Datasets and Pre-Processing

7.3.1 Evaluation Plan

In this section we outline the evaluation plan to verify the effectiveness of our learning approaches. To evaluate the news suggestion problem we are faced with two challenges.

- *What comprises the ground truth for such a task ?*
- *How do we construct training and test splits given that entity pages consists of text added at different points in time ?*

Consider the ground truth challenge. Evaluating if an arbitrary news article should be included in Wikipedia is both subjective and difficult for a human if she is not an expert. An invasive approach, which was proposed by Barzilay and Sauper [SB09], adds content directly to Wikipedia and expects the editors or other users to redact irrelevant content over a period of time. The limitations of such an evaluation technique is that content added to long-tail entities might not be evaluated by informed users or editors in the experiment time frame. It is hard to estimate how much time the added content should be left on the entity page. A more non-invasive approach could involve crowdsourcing of entity and news article pairs in an IR style relevance assessment setup. The problem of such an approach is again finding knowledgeable users or experts for long-tail entities. Thus the notion of *relevance* of a news recommendation is challenging to evaluate in a crowd setup.

We take a slightly different approach by making an assumption that the news articles already present in Wikipedia entity pages are relevant. To this extent, we extract a dataset comprising of all news articles referenced in entity pages (details in Section 7.3.2). At the expense of not evaluating the space comprising of news articles absent in Wikipedia, we succeed in (i) avoiding restrictive assumptions about the quality of human judgments, (ii) being invasive and polluting Wikipedia, and (iii) deriving a reusable test bed for quicker experimentation.

The second challenge of construction of training and test set separation is slightly easier and is addressed in Section 7.3.4.

7.3.2 Datasets

The datasets we use for our experimental evaluation are directly extracted from the Wikipedia entity pages and their revision history. The generated data represents one of the contributions of our work.[6] The datasets are the following:

Entity Classes. We focus on a manually predetermined set of *entity classes* for which we expect to have news coverage. The number of analyzed *entity classes* is 27, including 73, 734 entities with at least one news reference. The *entity classes* were selected from the DBpedia class ontology. Figure 7.3 shows the number of entities per class for the years (2009-2014).

News Articles. We extract all news references from the collected Wikipedia entity pages.[7] The extracted news references are associated with the sections in which they appear. In total there were 411, 673 news references, and after crawling we end up with 351, 982 successfully crawled news articles. The details of the news article distribution, and the number of entities and sections from which they are referred are shown in Table 7.3.

Article-Entity Ground-truth. The dataset comprises of the news and entity pairs $\langle n, e \rangle \rightarrow \{0, 1\}$. News-entity pairs are relevant if the news article is referenced in the entity page. Non-relevant pairs (i.e. negative training examples) consist of news articles that contain an entity but are not referenced in that entity's page. If a news article n is referred from e at year t, the features are computed taking into account the entity profiles at year $S_e(t-1)$.

[6] http://l3s.de/~fetahu/cikm2015/data/

[7] A news reference in Wikipedia is denoted by the template `{cite type='news' | url=''}`

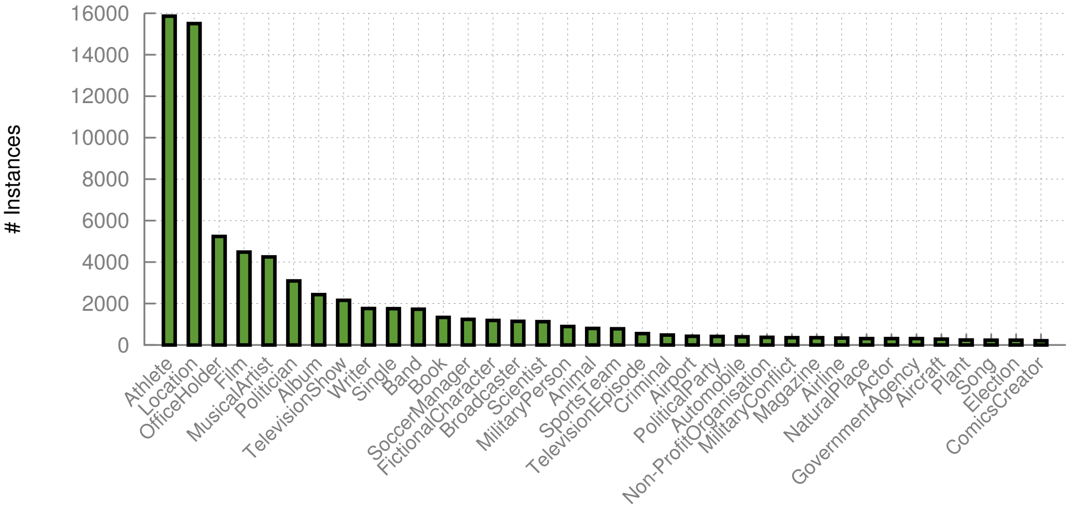


Figure 7.3. Number of entities with at least one news reference for different entity classes.

year	#news	#entities	#sections
2009	42707	13550	3510
2010	78328	24953	8416
2011	73491	23144	6581
2012	81473	25980	8455
2013	69079	22121	8183
2014	29961	11088	4694

Table 7.3. News articles, entities and sections distribution across years.

Article-Section Ground-truth. The dataset consists of the triple $\langle n, e, s\rangle$, where $s \in \widehat{S}_c$, where we assume that $\langle n, e\rangle$ has already been determined as relevant. We therefore have a multi-class classification problem where we need to determine the section of e where n is cited. Similar to the *article-entity* ground truth, here too the features compute the similarity between n, $S_e(t-1)$ and $\widehat{S}_c(t-1)$.

7.3.3 Data Pre-Processing

We POS-tag the news articles and entity profiles $S_e(t)$ with the Stanford tagger [TKMS03]. For entity linking the news articles, we use TagMe![FS12] with a confidence score of 0.3. On a manual inspection of a random sample of 1000 disambiguated entities, the accuracy is above 0.9. On average, the number of entities per news article is approximately 30. For entity linking the entity profiles, we simply follow the *anchor* text that refers to Wikipedia entities.

7.3.4 Train and Testing Evaluation Setup

We evaluate the generated supervised models for the two tasks, *AEP* and *ASP*, by splitting the train and testing instances. It is important to note that for the pairs $\langle n, e \rangle$ and the triple $\langle n, e, \widehat{S}_c \rangle$, the news article n is referenced at time t by entity e, while the features take into account the entity profile at time $t-1$. This avoids any 'overlapping' content between the news article and the entity page, which could affect the learning task of the functions $\mathcal{F}_e$ and $\mathcal{F}_s$. Table 7.4 shows the statistics of train and test instances. We learn the functions at year t and test on instances for the years greater than t. Please note that we do not show the performance for year 2014 as we do not have data for 2015 for evaluation.

	$\mathcal{F}_e$		$\mathcal{F}_s$	
	`train`	`test`	`train`	`test`
2009	74,005	469,386	19,399	218,757
2010	190,409	382,085	70,486	167,670
2011	286,588	292,398	115,286	122,870
2012	386,647	177,755	170,682	67,474
2013	471,209	59,172	218,538	19,618

Table 7.4. Number of instances for train and test in the *AEP* and *ASP* tasks.

7.4 Results and Discussion

7.4.1 Article–Entity Placement

Here we introduce the evaluation setup and analyze the results for the *article–entity (AEP)* placement task. We only report the evaluation metrics for the *'relevant'* news-entity pairs. A detailed explanation on why we focus on the *'relevant'* pairs is provided in Section 7.2.1.

Evaluation Setup

Baselines. We consider the following baselines for this task.

- **B1.** The first baseline uses only the salience-based features by Dunietz and Gillick [DG14].
- **B2.** The second baseline assigns the value *relevant* to a pair $\langle n, e \rangle$, if and only if e appears in the title of n.

Learning Models. We use *Random Forests* (RF) [Bre01]. We learn the RF on all computed features in Table 7.1. The optimization on RF is done by splitting the feature space into multiple trees that are considered as ensemble classifiers. Consequently, for each classifier it computes the margin function as a measure of the average count of predicting the correct class in contrast to any other class. The higher the margin score the more robust the model.

Metrics. We compute *precision* P, *recall* R and F1 score for the *relevant* class. For example, precision is the number of news-entity pairs we correctly labeled as relevant compared to our ground truth divided by the number of all news-entity pairs we labeled as relevant.

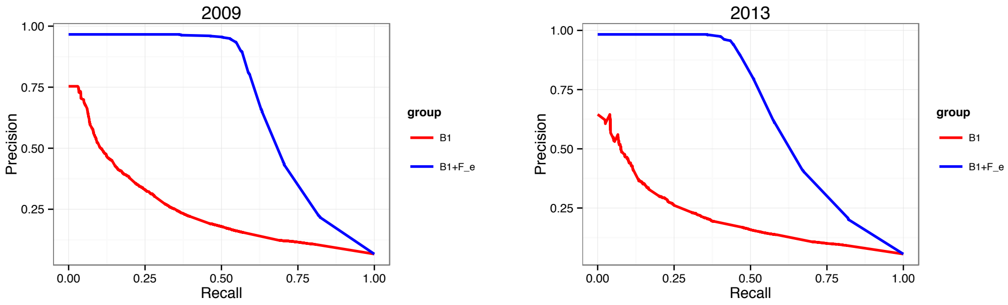
Approach Effectiveness

The following results measure the effectiveness of our approach in three main aspects: (i) overall *performance* of $\mathcal{F}_e$ and comparison to baselines, (ii) *robustness* across the years, and (iii) *optimal* model for the *AEP* placement task.

Performance. Figure 7.4 shows the results for the years 2009 and 2013, where we optimized the learning objective with instances from year t and evaluate on the years $t_i > t$ (see Section 7.3.4).[8] The results show the *precision–recall* curve. The *red* curve shows baseline **B1** [DG14], and the *blue* one shows the performance of $\mathcal{F}_e$. The curve shows for varying *confidence scores* (high to low) the precision on labeling the pair $\langle e, n \rangle$ as *'relevant'*. In addition, at each *confidence score* we can compute the corresponding recall for the *'relevant'* label. For high confidence scores on labeling the news-entity pairs, the baseline **B1** achieves on average a precision score of P=0.50, while $\mathcal{F}_e$ has P=0.93. We note that with the drop in the confidence score the corresponding precision and recall values drop too, and the overall F1 score for **B1** is around F1=0.2, in contrast we achieve an average score of F1=0.67.

It is evident from Figure 7.4 that for the years 2009 and 2013, $\mathcal{F}_e$ significantly outperforms the baseline **B1**. We measure the significance through the *t-test* statistic and get a *p-value* of $2.2e - 16$. The improvement we achieve over **B1** in absolute numbers, ΔP=+0.5 in terms of precision for the years between 2009 and 2014, and a similar improvement in terms of F1 score. The improvement for recall is Δ R=+0.4. The relative improvement over **B1** for P and F1 is almost 1.8 times better, while for recall we are 3.5 times better. In Table 7.5 we show the overall scores for the evaluation metrics for **B1** and $\mathcal{F}_e$. Finally, for **B2** we achieve much poorer performance, with average scores of P=0.21, R=0.20 and F1=0.21.

[8] We only show the first year 2009 and the last year 2013, since the difference to the other years is marginal.



(a) 2009 **(b)** 2013

Figure 7.4. Precision-Recall curve for the *article–entity* placement task, in *blue* is shown $\mathcal{F}_e$, and in *red* is the baseline **B1**.

Robustness. In Table 7.5, we show the overall performance for the years between 2009 and 2013. An interesting observation we make is that we have a very robust performance and the results are stable across the years. If we consider the experimental setup, where for year $t = 2009$ we optimize the learning objective with only 74k training instances and evaluate on the rest of the instances, it achieves a very good performance. We predict with F1=0.68 the remaining 469k instances for the years $t \in (2009, 2014]$.

The results are particularly promising considering the fact that the distribution between our two classes is highly skewed. On average the number of *'relevant'* pairs account for only around $4 - 6\%$ of all pairs. A good indicator to support such a statement is the *kappa* (denoted by κ) statistic. κ measures agreement between the algorithm and the gold standard on both labels while correcting for chance agreement (often expected due to extreme distributions). The κ scores for **B1** across the years is on average 0.19, while for $\mathcal{F}_e$ we achieve a score of 0.65 (the maximum score for κ is 1).

year	P		R		F1	
	B1	$\mathcal{F}_e$	**B1**	$\mathcal{F}_e$	**B1**	$\mathcal{F}_e$
2009	0.450	0.930	0.143	0.550	0.216	0.691
2010	0.503	0.939	0.128	0.540	0.204	0.685
2011	0.475	0.937	0.133	0.520	0.208	0.669
2012	0.476	0.935	0.110	0.515	0.177	0.664
2013	0.407	0.939	0.116	0.445	0.181	0.674

Table 7.5. *Article–Entity* placement task performance.

Feature Analysis

In Figure 7.5 we show the impact of the individual feature groups that contribute to the superior performance in comparison to the baselines. *Relative entity frequency* from the *salience* feature, models the entity salience as an exponentially decaying function based on the positional index of the paragraph where the entity appears. The performance of $\mathcal{F}_e$ with *relative entity frequency* from the *salience* feature group is close to that of all the features combined. The *authority* and *novelty* features account to a further improvement in terms of precision, by adding roughly a 7%-10% increase. However, if both feature groups are considered separately, they significantly outperform the baseline **B1**.

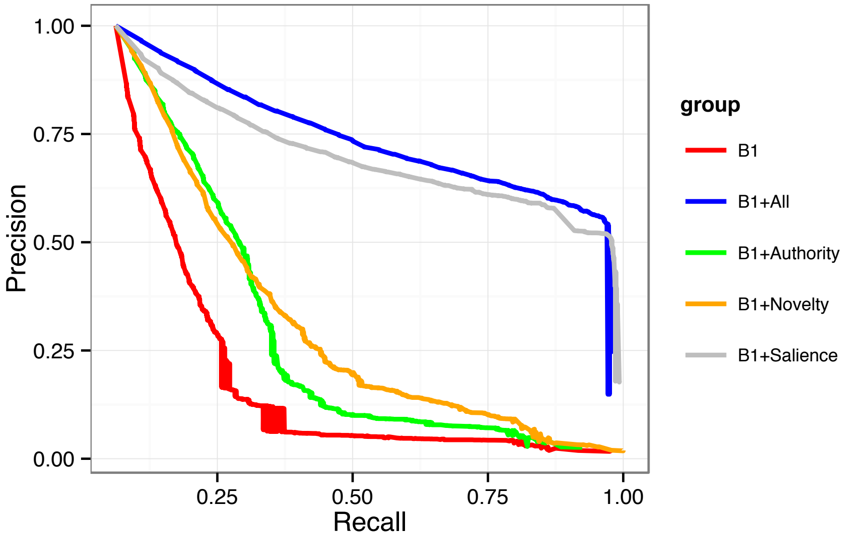


Figure 7.5. Feature analysis for the *AEP* placement task for $t = 2009$.

7.4.2 Article-Section Placement

Here we show the evaluation setup for *ASP* task and discuss the results with a focus on three main aspects, (i) the overall performance across the years, (ii) the *entity class* specific performance, and (iii) the impact on *entity profile* expansion by suggesting missing sections to entities based on the pre-computed templates.

Evaluation Setup

Baselines. To the best of our knowledge, we are not aware of any comparable approach for this task. Therefore, the baselines we consider are the following:

- **S1**: Pick the section from template $\widehat{S}_c$ with the highest lexical similarity to n: **S1**$=$ $\text{argmax}_{s\in\widehat{S}_c(t-1)}\langle n, e, s\rangle$
- **S2**: Place the news into the most frequent section in $\widehat{S}_c$

Learning Models. We use *Random Forests* (RF) [Bre01] and *Support Vector Machines* (SVM) [CL11]. The models are optimized taking into account the features in Table 7.2. In contrast to the *AEP* task, here the scale of the number of instances allows us to learn the SVM models. The SVM model is optimized using the $\epsilon - SVR$ *loss* function and uses the *Gaussian* kernels.

Metrics. We compute *precision* P as the ratio of news for which we pick a section s from $\widehat{S}_c$ and s conforms to the one in our ground-truth (see Section 7.3.2). The definition of *recall* R and F1 score follows from that of precision.

Overall Article-Section Performance

Figure 7.6 shows the overall performance and a comparison of our approach (when $\mathcal{F}_s$ is optimized using SVM) against the best performing baseline **S2**. With the increase in the number of training instances for the *ASP* task the performance is a monotonically non-decreasing function. For the year 2009, we optimize the learning objective of $\mathcal{F}_s$ with around 8% of the total instances, and evaluate on the rest. The performance on average is around P=0.66 across all classes. Even though for many classes the performance is already stable (as we will see in the next section), for some classes we improve further. If we take into account the years between 2010 and 2012, we have an increase of ΔP=0.17, with around 70% of instances used for training and the remainder for evaluation. For the remaining years the total improvement is ΔP=0.18 in contrast to the performance at year 2009.

On the other hand, the baseline **S1** has an average precision of P=0.12. The performance across the years varies slightly, with the year 2011 having the highest average precision of P=0.13. Always picking the most frequent section as in **S2**, as shown in Figure 7.6, results in an average precision of P=0.17, with a uniform distribution across the years.

Article-Section Performance per Entity Class

Here we show the performance of $\mathcal{F}_s$ decomposed for the different entity classes. Specifically we analyze the 27 classes in Figure 7.3. In Table 7.6, we show the results for a range of years (we omit showing all years due to space constraints). For illustration purposes only, we group them into four main classes ({ `Person, Organization, Location, Event`}) and into the specific sub-classes shown in

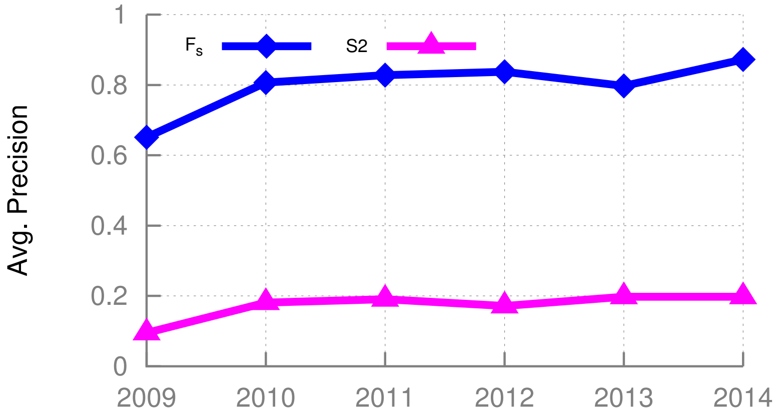


Figure 7.6. *Article-Section* performance averaged for all entity classes for $\mathcal{F}_s$ (using SVM) and **S2**.

the second column in Table 7.6. For instance, the entity classes `OfficeHolder` and `Politician` are aggregated into `Person–Politics`.

It is evident that in the first year the performance is lower in contrast to the later years. This is due to the fact that as we proceed, we can better generalize and accurately determine the correct *fit* of an article n into one of the sections from the pre-computed *templates* $\widehat{S}_c$. The results are already stable for the year range $(2009, 2012]$. For a few `Person` sub-classes, e.g. `Politics`, `Entertainment`, we achieve an F1 score above 0.9. These additionally represent classes with a sufficient number of training instances for the years $[2009, 2012]$. The lowest F1 score is for the `Criminal` and `Television` classes. However, this is directly correlated with the insufficient number of instances.

The baseline approaches for the *ASP* task perform poorly. **S1**, based on *lexical similarity*, has a varying performance for different entity classes. The best performance is achieved for the class `Person - Politics`, with P=0.43. This highlights the importance of our feature choice and that the *ASP* cannot be considered as a *linear function*, where the maximum similarity yields the best results. For different entity classes different features and combination of features is necessary. Considering that **S2** is the overall best performing baseline, through our approach $\mathcal{F}_s$ we have a significant improvement of over ΔP=+0.64.

The models we learn are very robust and obtain high accuracy, fulfilling our precondition for accurate news suggestions into the entity sections. We measure the robustness of $\mathcal{F}_s$ through the κ statistic. In this case, we have a model with roughly 10 labels (corresponding to the number of sections in a template $\widehat{S}_c$). The score we achieve shows that our model predicts with high confidence with $\kappa = 0.64$.

Entity class	Sub-Class	2009			(2009,2012]			(2012,2014]		
		P	R	F1	P	R	F1	P	R	F1
Person	Entert.	0.74	0.82	0.76	0.91	0.94	0.92	0.96	0.98	0.97
	Politics	0.92	0.94	0.93	0.92	0.95	0.93	0.94	0.96	0.95
	Scientists	0.48	0.68	0.55	0.89	0.94	0.91	0.93	0.95	0.94
	Sports	0.82	0.87	0.84	0.87	0.91	0.89	0.93	0.96	0.94
	Military	0.69	0.78	0.72	0.84	0.91	0.87	0.88	0.93	0.90
	Criminal	0.65	0.76	0.68	0.76	0.70	0.70	0.69	0.82	0.74
Organiz.	-	0.57	0.65	0.59	0.79	0.86	0.82	0.83	0.87	0.84
Creative Work	Television	0.53	0.65	0.56	0.75	0.73	0.71	0.73	0.77	0.75
	Music	0.60	0.62	0.59	0.86	0.75	0.76	0.90	0.94	0.91
	Written Work	0.66	0.77	0.70	0.73	0.83	0.77	0.72	0.79	0.74
Location	Location	0.78	0.76	0.72	0.86	0.90	0.87	0.92	0.96	0.94
Event	Event	0.56	0.68	0.61	0.86	0.87	0.85	0.69	0.72	0.69
	average	0.66	0.75	0.69	0.84	0.86	0.83	0.84	0.89	0.86

Table 7.6. *Article-Section* placement performance (SVM based $\mathcal{F}_s$) for the different *entity classes*.

Entity Profile Expansion

The last analysis is the impact we have on *expanding* entity profiles $S_e(t)$ with new sections. Figure 7.7 shows the ratio of sections for which we correctly suggest an article n to the right section in the section template $\widehat{S}_c(t)$. The ratio here corresponds to sections that are not present in the entity profile at year $t-1$, that is $s \notin S_e(t-1)$. However, given the generated templates $\widehat{S}_c(t-1)$, we can expand the entity profile $S_e(t-1)$ with a new section at time t. In details, in the absence of a section at time t, our model trains well on similar sections from the section template $\widehat{S}_c(t-1)$, hence we can predict accurately the section and in this case suggest its addition to the entity profile. With time, it is obvious that the expansion rate decreases at later years as the entity profiles become more 'complete'.

This is particularly interesting for expanding the entity profiles of long-tail entities as well as updating entities with real-world emerging events that are added constantly. In many cases such missing sections are present at one of the entities of the respective entity class c. An obvious case is the example taken in Section 7.2.1, where the *'Accidents'* is rather common for entities of type Airline. However, it is non-existent for some specific entity instances, i.e *Germanwings* airline.

Through our *ASP* approach $\mathcal{F}_s$, we are able to expand both *long-tail* and *trunk* entities. We distinguish between the two types of entities by simply measuring their section text length. The real distribution in the ground truth (see Section 7.3.2) is 27% and 73% are *long-tail* and *trunk* entities, respectively. We are able to expand the entity profiles for both cases and all entity classes without a significant difference, with the only exception being the class `Creative Work`, where we expand significantly more *trunk* entities.

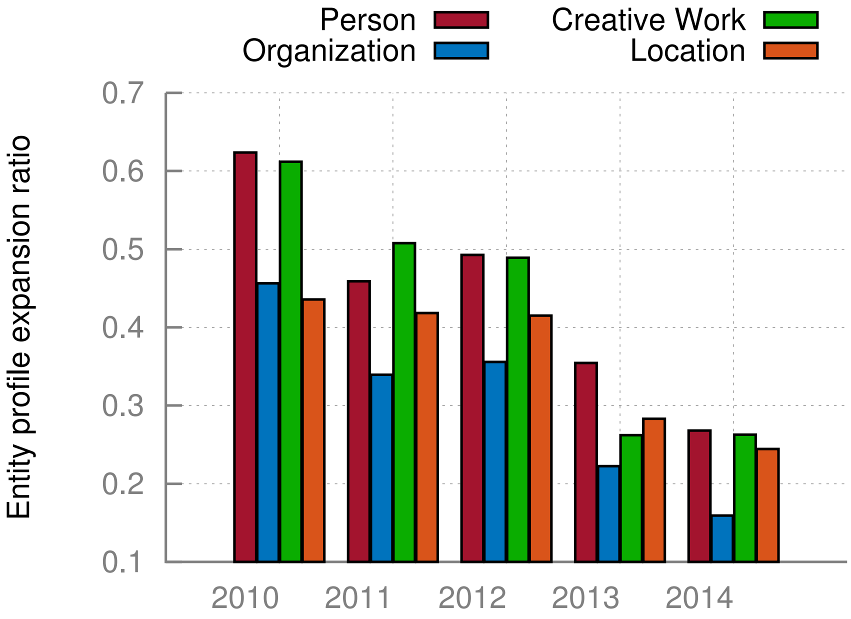


Figure 7.7. Correctly suggested news articles for $s \in S_e(t) \wedge s \notin S_e(t-1)$.

7.5 Conclusion

In this chapter, we have proposed an automated approach for the novel task of suggesting news articles to Wikipedia entity pages to facilitate Wikipedia updating. The process consists of two stages. In the first stage, *article–entity* placement, we suggest news articles to entity pages by considering three main factors, such as *entity salience* in a news article, *relative authority* and *novelty* of news articles for an entity page. In the second stage, *article–section* placement, we determine the best fitting section in an entity page. Here, we remedy the problem of incomplete entity section profiles by constructing section templates for specific entity classes. This allows us to add missing sections to entity pages. We carry out an extensive experimental evaluation on 351,983 news articles and 73,734 entities coming from 27 distinct entity classes. For the first stage, we achieve an overall performance with P=0.93, R=0.514 and F1=0.676, outperforming our baseline competitors significantly. For the second stage, we show that we can learn incrementally to determine the correct section for a news article based on section templates. The overall performance across different classes is P=0.844, R=0.885 and F1=0.860.

In the future, we will enhance our work by extracting facts from the suggested news articles. Results suggest that the news content cited in entity pages comes from the first paragraphs. However, challenging task such as the canonicalization and chronological ordering of facts, still remain.

Through this approach we are able to account for the constantly evolving nature of Wikipedia entities by suggesting important, novel, and authoritative information coming from news sources. The work in this chapter concludes the proposed approaches in this thesis, which account for a holistic framework of enriching and improving Wikipedia articles.

8

Entity Search as a Use Case of Wikipedia

In this chapter, we present the *entity search* use case, which is one of the most well-known use cases of Wikipedia. We present work we carried in this field, specifically on improving entity search from structured datasets.

Wikipedia as one of the largest online encyclopedias is an important source for generating structured datasets. As such it has been successfully used in various projects on generating large knowledge graphs like DBpedia [ABK+07], YAGO [SKW07].

Similar datasets, like DBpedia, referred to as Web data form highly heterogeneous knowledge-graphs with more than 100 billion triples [PMZ10]. Web data consist of a wide variety of languages, schemas, domains and topics [GGSL12]. Even though a large number of entities and concepts are highly overlapping, that is, they represent the same or related concepts, explicit links are still limited and often concentrated within large established knowledge graphs, like DBpedia.

The entity-centric nature of the Web of data has led to a shift towards tasks related to entity and object retrieval [BCMT13, TDCM12] or entity-driven text summarization [DMBZ10]. Major search engine providers such as Google and Yahoo! already exploit such data to facilitate semantic search using knowledge graphs, or as part of similar efforts such as the *EntityCube-Renlifang* project at Microsoft Research [NMS+07]. In such scenarios, data is aggregated from a range of sources calling for efficient means to search and retrieve entities in large data graphs.

Entity search [PMZ10, TDCM12] aims at retrieving relevant entities given a user query. The result is a ranked list of entities [BCMT13]. By simply applying standard keyword search algorithms, like the BM25F [BMV11, RZ09], promising results can be achieved. Other works rely on learning to rank approaches [ZA13], however, in the absence of training data, unsupervised retrieval models are preferred for ad-hoc queries.

In most cases, queries are entity centric. However, there are a large number of queries that are also topic-based, e.g. '`U.S. Presidents`'. Therefore, approaches like [TDCM12] have proposed retrieval techniques that make use of the explicit links between entities in the WoD for results or query expansion. For instance, following `owl:sameAs` or `rdfs:seeAlso` predicates from `dbp:Barack_Obama`, one can retrieve co-references or highly related entities. However, considering the size of the WoD such statements are very sparse (see Figure 8.1a).

We propose a method for improving entity retrieval in two aspects. We improve the task by *expanding* and *re-ranking* the result set from a baseline retrieval model (BM25F), and address the sparsity of explicit links between entities through clustering of entities based on their similarity, using a combination of lexical and structural features. Finally, we re-rank the expanded result set, based on how likely it is an entity and its type to be relevant for a user query, defined as *query type affinity*.

We experimentally evaluate on large structured dataset repository, namely the BTC12 dataset [Har12], and use the SemSearch[1] query dataset. The individual steps in our approach are evaluated through a reliable crowdsourced evaluation approach.

The main contributions of our work are as follows: (a) an entity retrieval model combining keyword search and entity clustering, and (b) an entity ranking model considering the query type affinity w.r.t the set of relevant entity types.

8.1 Approach and Overview

8.1.1 Preliminaries

The *entity search (ES) task* concerns with retrieving a top–k ranked set of entities from dataset for a given a user query q. User queries are typically entity centric. A *dataset* in our case is a set of triples $\langle s, p, o\rangle$ (see Chapter 2). An *entity* profile of e is the set of triples sharing the same subject URI s. The *entity type* is determined by the triple $t_e = \langle s, \texttt{rdf:type}, o\rangle$. Additionally, we define the *query type* t_q, corresponding to the entity type in q, e.g. *'Barack Obama'*, hence t_q `typeOf Person`.

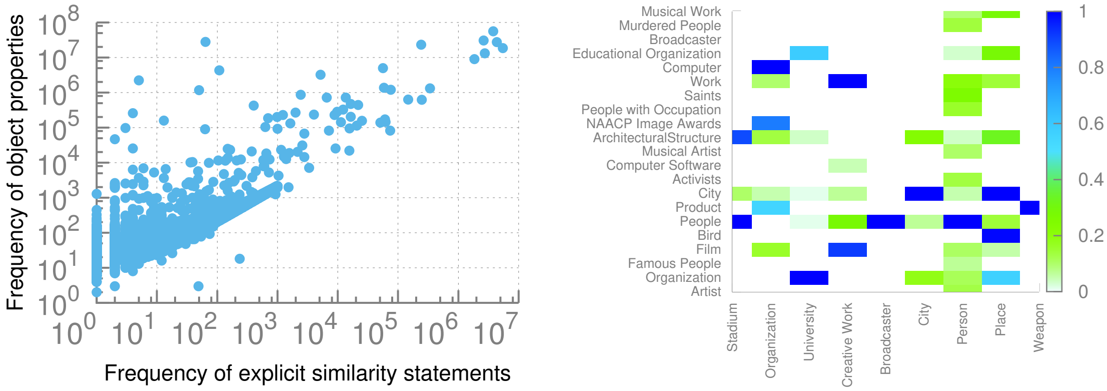
8.1.2 Motivation

Recent studies [TDCM12] have shown that *explicit similarity statements*, which indicate some form of similarity or equivalence between entities, e.g. `owl:sameAs`, are useful for improving entity search results. However, explicit similarity statements usually are sparse and often focused towards a few well established datasets like DBpedia, Freebase etc. One reason for this is that these datasets represent known, and well structured graphs, which show a comparably high proportion of similarity statements linking similar entities within and beyond their original namespace.

Figure 8.1a shows the total amount of explicit similarity statements (on the x–axis) that interlink entities in the BTC12 dataset. Referring to [TDCM12], here we specif-

[1] http://km.aifb.kit.edu/ws/semsearch10/

ically consider triples of the form $\langle e, p, e' \rangle$ where the predicate $p \in \{$`owl:sameAs`, `skos:related`, ..., `dbp:synonym`$\}$. These are plotted against the total number of *object properties* (y–axis), where each point in the plot represents a graph in the BTC12 collection. From the figure, it is obvious that the number of explicit similarity statements is very sparse, considering the size of the dataset.



(a)

(b)

Figure 8.1. (a) Number of explicit similarity statements in contrast to the frequency of object property statements overall, shown for all data graphs. (b) Query type affinity shows the query type and the corresponding entity types from the retrieved and relevant entities.

Nonetheless, missing links between entities can be partially remedied by computing their pair-wise similarity, thereby complementing statements like `owl:sameAs` or `skos:related`. Given the semi-structured nature of RDF data, graph-based and lexical features can be exploited for similarity computation. Particularly, lexical features derived from literals provided by predicates such as *rdfs:label* or *rdfs:description* are prevalent in LOD. Our analysis on the BTC12 dataset reveals that a large portion of entities (around 90%) have an average literal length of 50 characters.

Furthermore, while the query type usually is not considered in state of the art ES methods, we investigated its correlation with the corresponding entity types from the query result set. We refer to a ground truth[2] using the BTC10 dataset. We focus only on relevant entities for q. We analyze the *query type affinity* of the result sets by assessing the likelihood of an entity in the results to be of the same type as the query type. Figure 8.1b shows the query type affinity. On the x-axis we show the query type, whereas on the y-axis the corresponding relevant entity types are shown. Figure 8.1b shows that most queries have high affinity with a specific entity type, with the difference being the query type `Person`, where relevant entities have a wider range of types.

[2] http://km.aifb.uni-karlsruhe.de/ws/semsearch10/Files/assess

We exploit the *query type affinity* to improve the ranking of entities for a query q (see Section 8.3). Based on these observations, we argue that (a) *entity clustering* can remedy the lack of existing linking statements and (b) entity *re-ranking* considering the *query type affinity* are likely to improve the entity retrieval task.

8.1.3 Approach Overview

In this work we propose a novel approach for the *entity search* task which builds on the observations described earlier. Figure 8.2 shows an overview of the proposed approach. The individual steps are outlined below and described in detail in Section 8.2 and 8.3. We distinguish between two main steps: (I) *offline pre-processing*, including step I.a and I.b in the following overview, and (II) *online entity retrieval*, covered by steps II.a to II.c.

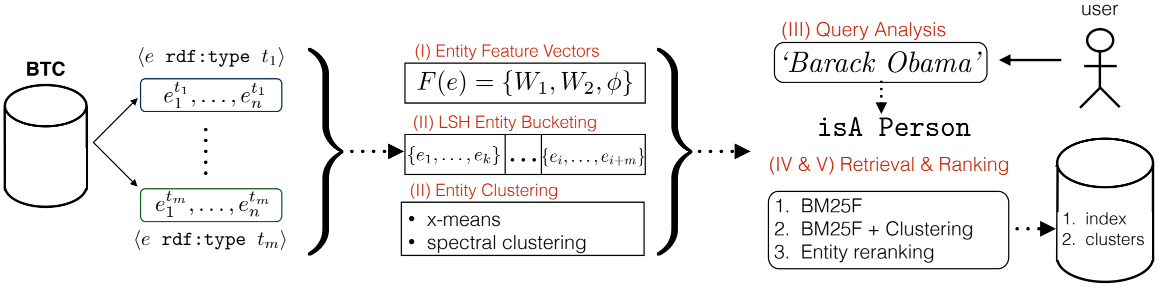

BTC
$\langle e$ rdf:type $t_1\rangle$
$e_1^{t_1}, \ldots, e_n^{t_1}$
$e_1^{t_m}, \ldots, e_n^{t_m}$
$\langle e$ rdf:type $t_m\rangle$
(I) Entity Feature Vectors
$F(e) = \{W_1, W_2, \phi\}$
(II) LSH Entity Bucketing
$\{e_1, \ldots, e_k\}$ … $\{e_i, \ldots, e_{i+m}\}$
(II) Entity Clustering
• x-means
• spectral clustering
(III) Query Analysis
'Barack Obama'
user
isA Person
(IV & V) Retrieval & Ranking
1. BM25F
2. BM25F + Clustering
3. Entity reranking
1. index
2. clusters

Figure 8.2. Overview of the *entity retrieval* approach.

8.2 Data Pre-processing and Entity Clustering

Here we describe the offline pre-processing to cluster entities and remedy the sparsity of explicit entity links.

8.2.1 Entity Feature Vectors

Entity similarity is measured based on a set of structural and lexical features, denoted by the *entity feature vector* $F(e)$. Finally, to avoid overfitting, we filter out items from $F(e)$ which have low frequency.

Lexical Features. We consider a weighted set of *unigrams* and *bigrams* for an entity e, by extracting all textual literals used to describe e denoted as $\mathbf{W}_1(e)$ and $\mathbf{W}_2(e)$. The weights are computed using the standard *tf–idf* metric. High lexical similarity between an entity pair is a good indicator for expanding the result set from the corresponding cluster space.

Structural Features. The feature set $\phi(e)$ considers the set of all object properties that describe e. The range of values for the structural features is $\phi(o, e) \rightarrow [0, 1]$, i.e., to indicate if a object value is present in e.

8.2.2 Entity Bucketing & Clustering

Entity Bucketing. In this step we *bucket* entities of a given *entity type* by computing their *MinHash* signature, which is used thereafter by the LSH algorithm [RU11]. This step is necessary as the number of entities is very large. In this way we reduce the number of pair-wise comparisons for the entity clustering, and limit it to only the set of entities within a bucket. Depending on the *clustering algorithm*, the impact of bucketing on the clustering scalability varies. Since the LSH algorithm itself has linear complexity, bucketing entities presents a scalable approach considering the size of datasets in our experimental evaluation.

Entity Clustering. We cluster entities separately for the different entity types and the computed LSH buckets. We consider two clustering approaches: (i) *X–means* and (ii) *Spectral Clustering*. In both approaches we use Euclidean distance as the similarity metric. The dimensions of the Euclidean distance are the feature items in $F(\cdot)$. The similarity metric is defined in Equation 8.1, whereas the details of the two different clustering algorithms are explained in Chapter 2.

$$d(e, e') = \sqrt{\sum \left(\mathbf{F}(e) - \mathbf{F}(e')\right)^2} \tag{8.1}$$

where the sum aggregates over the union of feature items from $\mathbf{F}(e), \mathbf{F}(e')$. The outcome of this process is a set of clusters $\mathbf{C} = \{C_1, \ldots, C_n\}$. The clustering process represents a core part of our approach from which we expand the entity results set for a given query, beyond the entities that are retrieved by a baseline as a starting point. The way the clusters are computed has an impact on the entity retrieval task, thus we present a thorough evaluation of cluster configurations in Section 8.5.1.

8.3 Entity Search: Expansion and Ranking

Result-set Expansion. From the initial result set $E_b = \{e_1, \ldots, e_k\}$ we obtain through BM25F, we expand the result set. From entities in E_b, we extract their corresponding set of clusters $\mathbf{C}$ as computed in the pre-processing stage. The result set is expanded with entities belonging to the clusters in $\mathbf{C}$. We denote the entities extracted from the clusters with E_c.

There are several precautions that we need to take into account in this step. We define two threshold parameters for expanding the result set. First, *cluster size*, where for a number of entities above a threshold in a cluster we do not take the entities into account entities. The rationale is that clusters with a large number of entities tend to be less homogeneous, i.e. they tend to be a weak indicator of similarity. The second parameter deals with the *number of entities* with which we expand the result set for a given entity cluster. The entities are considered based on their distance to the entity e_b. We experimentally validate the two parameters in Section 8.5.

We measure the similarity of entities in E_c w.r.t query q in Equation 8.2, where the φ measures *string* distance of e_c to q. Furthermore, this is done relative to entity e_b, such that if the e_b is more similar to q, $\varphi(q, e_b) < \varphi(q, e_c)$ the similarity score will be increased, hence, the expanded entity e_c will be penalized later on in the ranking. The second component represents the actual distance score $d(e_b, e_c)$.

$$sim(q, e_c) = \lambda \frac{\varphi(q, e_c)}{\varphi(q, e_b)} + (1 - \lambda) d(e_b, e_c) \tag{8.2}$$

we set $\lambda = 0.5$, such that entities are scored equally with respect to their match to query q and the distance between entities. In this step we identify possibly relevant entities that have been missed by the scoring function of BM25F. Such entities could be suggested as relevant from the extensive clustering approaches that consider the structural and lexical similarity.

Query Analysis. Following the example in Figure 8.1b, an important factor on the ranking of the result set is the *query type affinity*. It models the relevance likelihood of a given entity type t_e for a specific query type t_q. We give priority to entities that are likely to be relevant to the query type t_q and are least likely to be relevant for other query types t'_q. The probability distribution is modeled empirically based on a previous dataset, BTC10. The score γ, we assign to any entity coming from the expanded result set is computed as in Equation 8.3.

$$\gamma(t_e, t_q) = \frac{p(t_e|t_q)}{\sum\limits_{t'_q \neq t_q} \big(1 - p(t_e|t'_q)\big)} \tag{8.3}$$

An additional factor we use in the re-ranking process is the *context score*. To better understand the query intent, we decompose a query q into its *named entities* and additional *contextual terms*. An example is the query $q = \{$*'harry potter movie'*$\}$ from our query set, in which case the contextual terms would be '*movie*' and the named entity '*Harry Potter*' respectively. In case of ambiguous queries, the contextual terms can further help to determine the query intent. The *context score* (see Equation 8.4) indicates the relevance of entity e to the contextual terms Cx of the query q. For entities with a high number of textual literals, we focus on the main literals like *labels, name* etc.

$$context(q, e) = \frac{1}{|Cx|} \sum_{c_x \in Cx} \mathbb{1}_{e \texttt{ has } c_x} \tag{8.4}$$

Ranking In the final step, we rank the expanded entity result set. The result set is the union of entities $\mathbf{E} = \mathbf{E}_b \cup \mathbf{E}_c$. In the case of entities retrieved through the baseline approach $e \in \mathbf{E}_b$, we simply re-use the original score, but normalize the values between $[0, 1]$. For entities from E_c we normalize the similarity score relative to the rank of entity e_b (the position of e_b in the result set) which was used to suggest e_c. This boosts entities which are the result of expanding top-ranked entities.

$$rank\_score(e) = \begin{cases} \frac{sim(q,e)}{rank(e_b)} & \text{if } e \in E_c \\ bm25f(q,e) & \text{otherwise} \end{cases} \tag{8.5}$$

The final ranking score $\alpha(e, t_q)$, for entity e and query type t_q assigns higher rank score in case the entity has high similarity with q and its type has high relevance likelihood of being relevant for query type t_q. Finally, depending on the query set, in case q contains contextual terms we can add $context(q, e)$ by controlling the weight of λ (in this case $\lambda = 0.5$).

$$\alpha(e, t_q) = \lambda\,(rank\_score(e) * \gamma(t_e, t_q)) + (1 - \lambda) * context(q, e) \tag{8.6}$$

8.4 Experimental Setup

Here we describe our experimental setup, specifically the datasets, baselines and the ground truth. The setup and evaluation data are available for download[3].

8.4.1 Evaluation Data

Dataset. We use the BTC12 dataset [Har12]. It represents one of the largest periodic crawls of Linked Data, also containing well-known knowledge bases like Freebase and DBpedia. The overall statistics of the data are: (i) 1.4 billion triples, (ii) 107,967 graphs, (iii) 3,321 entity types, and (iv) 454 million entities.

Entity Clusters. The statistics for the generated clusters are as follows: the average number of entities fed into the *LSH* bucketing algorithm is 77,485, whereas the average number of entities fed into *x–means* and *spectral* is 400. The number of generated entity buckets by LSH is 20,2009, while the number of clusters for *x–means* and *spectral* is 13 and 38, with an average of 10 and 20 entities per cluster respectively.

Query Dataset. We use *SemSearch*[4] query set from 2010 with 92 queries. The SemSearch query set is a standard collection for evaluating entity retrieval tasks.

[3] http://l3s.de/~fetahu/iswc2015/

[4] http://paragraph.aifb.kit.edu/ws/semsearch10/

8.4.2 Baseline and State of the Art

Baseline. We distinguish between two cases for the original BM25F baseline: (i) $\mathbf{B_t}$ and (ii) $\mathbf{B_b}$. In the first case, we use the *title* or *label* of an entity as a query field, whereas in the second case we use the full *body* of an entity (consisting of all textual literals). The scoring of the fields is performed similar as in [BMV11].

State of the art. We consider the approach proposed in [TDCM12] as the state-of-the-art. Similar to their experimental setup, we analyze two cases: (i) **S1** and (ii) **S2**. **S1** expands the entity set from the baseline approach with directly connected entities, and **S2** expands with entities up to the second hop. For further details we refer the reader to [TDCM12]. In our experiments, we found that the **S2** did not result in any significant change in performance when compared to **S1**, and we therefore do not report further on **S2**.

Our approaches. We analyze two entity retrieval techniques from our approach. The first is based on the *x–means* clustering approach, which we denote by **XM**. The second technique is based on *spectral* clustering and is denoted by **SP**. In both cases, we only expand the result set with entities coming from clusters with a total of ten entities associated with a cluster, and finally add only the most relevant entity based on the $sim(q, e_c)$ score.

BTC indexes. For the baseline, we generate a Lucene index, where we index entity profiles on two fields `title` and `body` (consisting of all the textual literals of an entity). The second index is an RDF index over the BTC dataset with support for SPARQL queries, for which we use the RDF3X tool [NW08]. The first index is used for the baseline approach, while the second for the state of the art approach.

8.4.3 Ground Truth for Evaluation of Entity Retrieval

For each query, we establish the ground-truth of relevant entities from the result-set through crowdsourcing. Crowdsourced evaluation campaigns for the task of ad-hoc object retrieval have been shown to be reliable [BHH+11, HHM+10]. For each of the 92 queries, we pool the top 50 entities retrieved by the various methods, resulting in the top-k pooled entities corresponding to the query. By doing so we generate 4,600 query-entity pairs.

We follow the key prescriptions for task design and deployment that emerged from the work of Blanco et al. [BHH+11] to build a ground truth. Workers are asked to assess the relevance of each retrieved entity to the corresponding query on a 5-point Likert-type scale[5].

[5] *1:Not Rel.*, *2:Slightly Rel.*, *3:Moderately Rel.*, *4:Fairly Rel.* and *5:Highly Rel.*

We collect 5 judgements from different workers for each pair to ensure reliable relevance assessments and discernible agreement between workers. This results in a total of 23,000 judgements. The final relevance of an entity is considered to be the aggregated relevance score over the 5 judgements. We assess and compare the performance of the different methods by relying on the ground truth thus generated (see Section 8.5).

8.4.4 Evaluation Metrics

Evaluation metrics assess the clustering accuracy and the retrieval performance. We indicate with e_i^{rel} a retrieved relevant entity retrieved at rank i.

Cluster Accuracy. As an initial evaluation, we assess the quality of our clusters. From a set of entities belonging to the same cluster, the accuracy is measured as the ratio of entities that *belong together* over the total number of entities in a cluster, where assessments are obtained through crowdsourcing (see Section 8.5).

Precision. P@k measures the precision at rank k, in our case $k = \{1, \dots, 10\}$. It is measured as the ratio of retrieved and relevant entities up to rank k over the total number of entities retrieved up to rank k.

$$P@k = \frac{1}{k} \sum_{i=1}^{k} e_i^{rel} \tag{8.7}$$

Recall. R@k is measured as the ratio of retrieved and relevant entities up to rank k over the total number of relevant entities up to rank k, indicated by $|e_k^{rel}|$ in the equation below.

$$R@k = \frac{1}{|e_k^{rel}|} \sum_{i=1}^{k} e_i^{rel} \tag{8.8}$$

Mean Average Precision . MAP provides an overall precision of a retrieval approach across all queries under consideration.

$$MAP = \frac{1}{|Q|} \sum_{q \in Q} P@k(q) \tag{8.9}$$

Normalized Discounted Cumulative Gain . It takes into account the ranking of entities generated using one of the retrieval approaches and compares it against the ideal ranking in the *ground truth*.

$$nDCG@k = \frac{DCG@k}{iDCG@k} \quad DCG@k = rel_1 + \sum_{i=2}^{k} \frac{rel_i}{log_2 i}$$

where $DCG@k$ represents the discounted cumulative gain at rank k, and $iDCG@k$ is the ideal $DCG@k$ computed from the *ground truth*.

8.5 Evaluation and Discussion

In this section we report evaluation results of the two main steps in our approach. We first evaluate the quality of the pre-processing step, i.e., the clustering results for the *x–means* and *spectral* clustering algorithms. Next, we present the findings from our rigorous evaluation of the entity retrieval task.

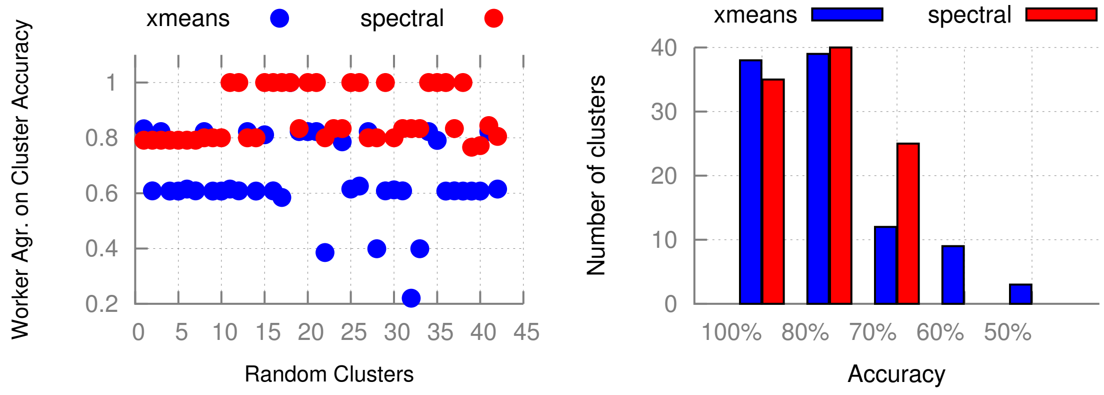
8.5.1 Cluster Accuracy Evaluation

Considering the large number of clusters that are produced in the pre-processing step for a *type* and *bucket*, evaluating the accuracy and quality of all clusters is infeasible. Thus, we randomly pick 10 entity types and 10 buckets, resulting in 100 clusters for evaluation. For each cluster, we randomly select a maximum of 10 entities.

To evaluate the *cluster accuracy*, we deploy atomic microtasks modeled such that a worker is presented with sets of 10 entities belonging to a cluster, along with a description of the entity in the form of the entity profile. The task of the worker is to pick the odd entities out (if any). We gather 5 judgments from different workers for each cluster. By enforcing restrictions available on the CrowdFlower platform, and following state of the art task design recommendations, we ensure that we receive judgments from the best workers (workers with high reputation as indicated by CrowdFlower).

Figure 8.3b presents our findings for the evaluation of the clustering process. We note that for *x–means* and *spectral* clustering approaches, nearly 35% and 38% of the clusters are judged to be perfect respectively (i.e., the entities within the cluster were all found to belong together). 39% of the clusters corresponding to *spectral* clustering and 40% of the clusters corresponding to *x-means*, have an accuracy of 80%. Considering its multidimensional representation of the entities, *spectral* clustering has higher accuracy and it does not have clusters below 70% accuracy. The lowest accuracy of 70% for *spectral* clustering implies that in each cluster there were only 3 entities that did not belong to the cluster. The implications of an accurate clustering process become clearer in the next section, where we assess the accuracy of finding relevant entities in the generated entity clusters.

Figure 8.3a presents the pairwise agreement between workers on the quality of each cluster. In case of the *spectral* clustering, we observe a high inter-worker agreement



(a) (b)

Figure 8.3. (a) Worker agreement on cluster accuracy for *spectral* and *x–means* clustering. (b) Cluster accuracy for the *spectral* and *x–means* clustering approaches.

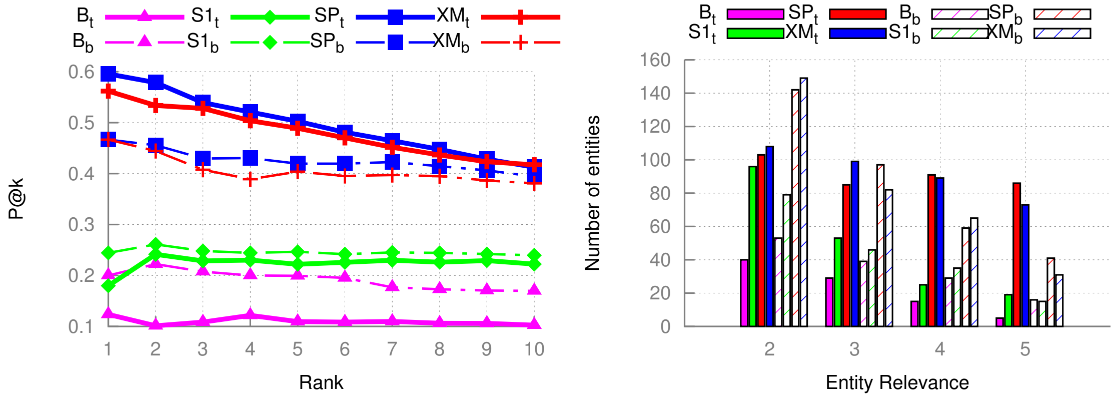
of 0.75 as per Krippendorf's Alpha. We observe a moderate inter-worker agreement of 0.6 as per Krippendorf's Alpha on the clusters resulting from *x–means*.

8.5.2 Entity Retrieval Evaluation

Figure 8.4a presents a detailed comparison between the $P@k$ for the different methods. The proposed approaches outperform the baseline and state of the art at all ranks. The precision is highest at $P@1 = 0.6$ whereas for the later ranks it stabilizes at 0.4. In contrast to our approach, the performance of the baseline and the state of the art is more uniform, and is around $P@k = 0.25$. The best overall performing approach is the retrieval approach based on spectral clustering SP. Table 8.1 shows the details about the performance of the respective approaches as measured for our evaluation metrics.

An interesting observation is that for our approaches the best performance is achieved when querying for the field *title*. In the case of the baseline, the best performance is achieved when querying for the field *body* (B_b) while the same is inconclusive in case of the state-of-the-art methods ($S1_t$ and $S1_b$). We achieve a significantly higher retrieval performance when using the title field. This can be explained by the fact that entities that match a query on their *title* field when compared to those that match a query on their *body* field, have a higher likelihood of being an exact match.

The high gain in performance through our methods (SP and XM) stems mainly from the two steps in our approach. The first step expands the result set with relevant entities as shown in Figure 8.4b. The figure shows the number of relevant entities corresponding to the different grading scales as described in Section 8.5.1. In all cases we note that our methods find more relevant entities. The second step which re-ranks



(a) **(b)**

Figure 8.4. (a) P@k for the different entity retrieval approaches under comparison. (b) The relevant entity frequency based on their graded relevance (from *2-Slightly Relevant* to *5-Highly Relevant*) for the different methods.

the expanded result set helps in reducing the number of *'non-relevant'* entities. We find that $S1_t$ has a 14% decrease of non-relevant entities, whereas SP_t and XM_t depict a 35% decrease, respectively. In second case where we query the *body* field, the number of *'non-relevant'* entities for $S1_b$ decreases by about 13%, while SP_b and XM_b depict a 24% decrease.

	B_t	B_b	$S1_t$	$S1_b$	SP_t	SP_b	XM_t	XM_b
P@10	0.103	0.170	0.222	0.240	0.413	0.394	**0.417**	0.381
R@10	0.052	0.089	0.112	0.118	0.206	**0.219**	0.216	0.215
MAP	0.110	0.191	0.224	0.246	**0.497**	0.426	0.482	0.407
$Avg(R)$	0.031	0.058	0.063	0.074	0.132	**0.133**	0.131	0.130

Table 8.1. Performance of the different entity retrieval approaches. In all cases our approaches are significantly better in terms of P/R ($p < 0.05$ measured for *t-test*) compared to *baseline* and *state of the art*. There is no significant difference between SP and XM approaches.

We additionally analyze the performance of the entity retrieval approaches through the $NDCG@k$ metric. Figure 8.5 shows the NDCG scores. Similar to our findings for $P@k$ presented in Table 1, our approaches perform best for the query field *title* and significantly outperform the approaches under comparison.

Next, we present observations concerning the different *query types* and the entity result set expansion parameters. In Figure 8.6a we show the improvement we gain in terms of MAP for the different *query types*. We observe that there is quite a variance for the different query types, however, in nearly all cases, the biggest improvement is achieved through the SP approach. Interestingly for the query type *'Creative Work'* the state of the art is nearly as good as the XM approach, whereas in the case of *'Weapon'* the baseline performs best. One possible explanation for this is that in the case of *'Creative Work'* the explicit entity similarity statements are abundant.

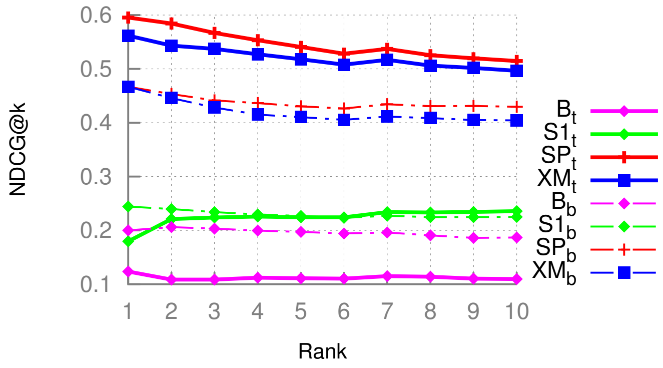
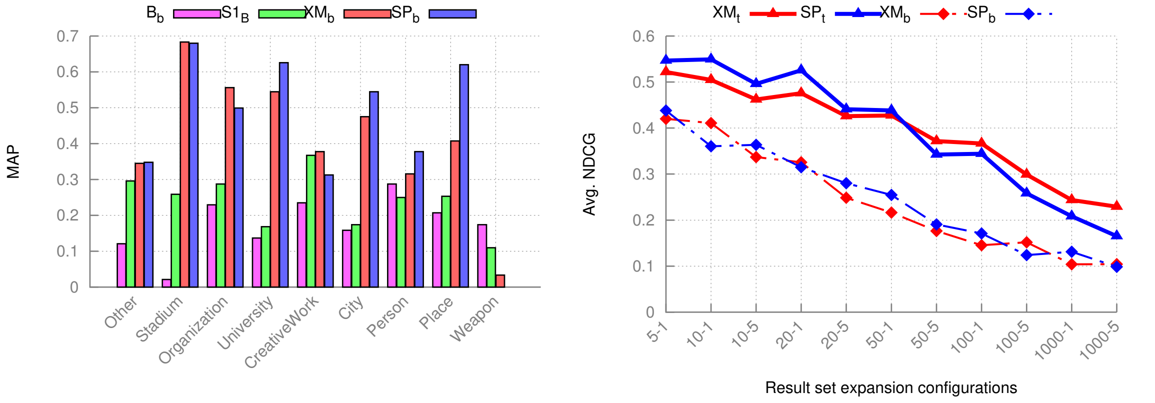


Figure 8.5. NDCG@k for $B1$, $S1$ and SP, XM

Addressing the case of optimizing our retrieval approaches, SP and XM, we experimentally show the impact that the expansion of the result set has on the measured performance metrics. Here, we show the impact on the *average NDCG* score. Figure 8.6b shows the performance at average NDCG for the varying *cluster size* and *number of entities* added (result set expansion) for every entity in E_b. The best performance is achieved for a rather smaller *cluster size* ranging between 5 and 10 entities per cluster. Regarding the number of entities with which the result set is expanded for every e_b, the best performance is achieved by expanding with one entity per cluster. The increase in cluster size and number of entities attributes to a decrease in performance.

8.6 Conclusions

In this work, we presented an approach to improve the performance of entity retrieval on structured data. Building on existing state of the art methods, we follow an approach consisting of offline preprocessing clustering, and online retrieval, results expansion and reranking. Preprocessing exploits *x–means* and *spectral* clustering algorithms using lexical as well as structural features. The clustering process was carried out on a large set of entities (over 450 million). The evaluation of the clustering process shows that over 80% of clusters have an accuracy of more than 80%. As part of the online entity retrieval, for a given a starting result set of entities as retrieved by the baseline approach BM25F we further expand the result set with relevant entities. Additionally, we propose an entity ranking model that takes into account the query type affinity. Finally, we carry out an extensive evaluation of the retrieval process using the SemSearch and the BTC12 datasets. The results show that our methods



(a) **(b)**

Figure 8.6. (a) The aggregated MAP for different query types and for the different retrieval approaches (note, we show the results for field *body* where baseline performs best). (b) The various configurations for the number of expanded entities for SP and XM.

outperform the baseline and state of the art approaches. In terms of standard IR metrics, our method in combination with one of the clustering approaches, e.g. SP_t improves over $S1_t$ with $\Delta P@10 = +0.19$, $\Delta MAP = +0.273$ and $\Delta R@10 = +0.1$.

9 Conclusion and Future Work

9.1 Conclusion and Contributions

The wide popularity of Wikipedia, with an ever growing number of real-world entities and events, has resulted in Wikipedia establishing itself as the reference point for a large number of Web users, and currently is ranked in the top–5 of the most visited internet sites [Sit]. Furthermore, due to its wide coverage it has found use in a large range of applications such as Web search [DGH+14, PMZ10], Apple's Siri, etc., and as such indirectly impacts a larger audience of users. One argumentation for its popularity, is that Wikipedia, represents an open and collaboratively edited encyclopedia, consisting of information and facts for real-world entities. For popular entities and events, it was found that emerging facts are reflected with a minimal delay in Wikipedia [KGC11].

To better understand such dynamics, on how emerging facts and information about entities are reflected in Wikipedia, we conducted a study in Chapter 2. We analyzed the amount of time it takes when something is reported in online news media, until it is reflected in Wikipedia. Additionally, we analyzed one of the key policies of providing citations to external sources as a means of evidence for added information by the Wikipedia editors. Not surprisingly, we found out that there is a long-tail of entities for which there is large gap between the time something is reported in news and until it is added in Wikipedia. Furthermore, we found that, the second most cited source in Wikipedia refers to online news media.

In this thesis we presented a holistic approach which has impact on several aspects of Wikipedia. The main outcomes and conclusions can be summarized as following:

- In Chapter 5, we enforce the *verifiability* principle of Wikipedia, where we provide evidence for statements that require citations in Wikipedia articles, specifically we focus on *news citations*. Hence, through the *verifiability* principle we improve the overall quality of Wikipedia articles. The main conclusions in this Chapter 5 are the following:

1. In order to find news citations for statements in Wikipedia articles, we first categorize the statements into the different citation categories (e.g. *web, news, journal* etc.). Through this step we can focus on specific categories for which we are in hold of such specific-collections (i.e. news collections). Furthermore this ensures that we suggest most appropriate citations. The accuracy of this task varies for the different entity types, and we achieve results with up to 80% on determining the appropriate citation type for Wikipedia statements.
2. After having determined the required citation type for a statement, next, we focused on only statements that require news citations. For such statements, it is important that the suggested news articles, entail the statement and preferably are central to the news article. We proposed a combined approach, which first retrieves a top–k set of news article candidates, and is followed by a binary classification, which tags the news article candidates as appropriate citations or not for the given statement. For this task, we achieve highly accurate results with more than 86% accuracy. Finally, we show that in 19% of the cases, we are able to suggest news citations of higher quality than those already existing in Wikipedia.

- In Chapter 6 we dealt with the problem of determining the span of citations in Wikipedia. This is an important aspect as citations in Wikipedia are assigned at various statement granularities (e.g. *sub-sentence, sentence* or *paragraph*), and as such there are no explicit means on knowing the exact span. This has direct consequences on enforcing the *verifiability* principle, where on the presence of a citation in a Wikipedia paragraph, it is not possible to differentiate the textual fragments that are covered or uncovered by a citation.

  We proposed an approach which relies on a *linear-chain* CRF for determining the citation span for *web* and *news* references. We achieve an overall accuracy of 90% across the different granularities of citation span.

- The work in Chapter 7 deals with the problem of Wikipedia's *coverage* of novel and important facts, and provides means on keeping up-to-date *long* and *trunk* Wikipedia articles. Since the nature of Wikipedia articles is constantly evolving, new information emerges and thus such changes need to be reflected into the structure of the Wikipedia articles themselves. We proposed a two-staged approach, which focuses on news sources for providing important and novel facts for Wikipedia articles.

  1. In the first stage, for a given news article, we analyze its content and the entities that are mentioned therein. We first, compute entity salience measures, determining that the mentioned entities are central concept in the given article. In this way, we suggest to Wikipedia articles, only those news articles where the corresponding entity mention (i.e. thus referring

to a specific Wikipedia page) is important. We proposed novel salience measures which show significant improvement over existing state of the art approaches.

2. After determining for a Wikipedia article a set of news articles which provide important and novel information, next, based on a section template which we construct for specific types of Wikipedia articles, we determine the exact section for which we suggest such news articles. In case, a section is missing, we add the appropriate section automatically from the constructed templates.

Lastly, in Chapter 8, we showed an application use-case of Wikipedia as the main source for generating structured datasets, and proposed means on improving existing *entity search* approaches over structured datasets.

9.2 Future Work Directions

While we address several major issues with enriching and improving Wikipedia articles, there are several issues that still need to be addressed. In this thesis, through our contributions we enforce the *verifiability* principle by providing citations for Wikipedia text, and correspondingly determining explicitly the span of such citations, however, we are limited to only citation markers that are already present in Wikipedia. Furthermore, an already well known problem of Wikipedia edit wars leads to content on controversial being biased [DLM13].

Therefore, as future directions we foresee work on automatically placing citation markers, in case such markers do not exist already and are necessary. We plan to use the existing markers as ground-truth for automated approaches on placing citation markers for long-tail entities. This problem is in particular challenging, as we need to determine the type of the citation marker, that is, to what citation category the reference should point at. Alternatively, one could employ the proposed approach in Chapter 5 to determine for a piece of text that we want to place the citation marker which citation category is more suitable.

Next, for controversial or polarizing topics, collaborative editing or discussion lead to a phenomena known as *"echo chambers"*. As such communities that tend to agree on a specific matter are isolated from other communities who have a different stance [VBZ+15]. Such controversial behavior tends to be diffused in Wikipedia too, therefore, leading to Wikipedia edit wars, an in other cases to biased content towards a particular group on a controversial topic [DLM13]. To avoid biased content, we plan on analyzing the edits in the revision history of a Wikipedia article and the editor behavior to flag biased content, which may violate the neutrality of an article, respectively the *neutral point of view* policy [Hub17].

Bibliography

[ABK$^+$07] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th ISWC*, pages 722–735, 2007.

[AR12] Amjad Abu-Jbara and Dragomir R. Radev. Reference scope identification in citing sentences. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 3-8, 2012, Montréal, Canada*, pages 80–90, 2012.

[Arc] Internet Archive. Internet archive. https://archive.org/index.php.

[ASL12] Maik Anderka, Benno Stein, and Nedim Lipka. Predicting quality flaws in user-generated content: the case of wikipedia. In *The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012*, pages 981–990, 2012.

[AVR] Gianni Amati and Cornelis Joost Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20.

[BCHP07] Axel Bernal, Koby Crammer, Artemis Hatzigeorgiou, and Fernando Pereira. Global discriminative learning for higher-accuracy computational gene prediction. *PLoS Comput Biol*, 3(3):e54, 2007.

[BCMT13] Roi Blanco, Berkant Barla Cambazoglu, Peter Mika, and Nicolas Torzec. Entity recommendations in web search. In *Proceedings of the 12th ISWC*, pages 33–48, 2013.

[BF98] Paul S. Bradley and Usama M. Fayyad. Refining initial points for k-means clustering. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998), Madison, Wisconsin, USA, July 24-27, 1998*, pages 91–99, 1998.

[BFOS84] Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

[BHH$^+$11] Roi Blanco, Harry Halpin, Daniel M. Herzig, Peter Mika, Jeffrey Pound, Henry S. Thompson, and Duc Thanh Tran. Repeatable and reliable search system evaluation using crowdsourcing. In *Proceeding of the 34th ACM SIGIR*, pages 923–932, 2011.

[BIA14] Judit Bar-Ilan and Noa Aharony. Twelve years of wikipedia research. In *Proceedings of the 2014 ACM Conference on Web Science*, WebSci '14, pages 243–244, New York, NY, USA, 2014. ACM.

[Bib91] Douglas Biber. *Variation across speech and writing*. Cambridge University Press, 1991.

[Bis06] Christopher M Bishop. Pattern recognition. *Machine Learning*, 128:1–58, 2006.

[BLHL$^+$01] Tim Berners-Lee, James Hendler, Ora Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.

[BLK$^+$09] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. Dbpedia - A crystallization point for the web of data. *J. Web Sem.*, 7(3):154–165, 2009.

[BMV11] Roi Blanco, Peter Mika, and Sebastiano Vigna. Effective and efficient entity search in rdf data. In *Proceedings of the 10th ISWC*, pages 83–97, Berlin, Heidelberg, 2011. Springer-Verlag.

[BNJ03] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, 2003.

[BR13] Krisztian Balog and Heri Ramampiaro. Cumulative citation recommendation: classification vs. ranking. In *The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '13, Dublin, Ireland - July 28 - August 01, 2013*, pages 941–944, 2013.

[Bre01] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[Bri] Britannica. Encyclopedia britannica.

[BRTN13] Krisztian Balog, Heri Ramampiaro, Naimdjon Takhirov, and Kjetil Nørvåg. Multi-step classification approaches to cumulative citation recommendation. In *Open research Areas in Information Retrieval, OAIR '13, Lisbon, Portugal, May 15-17, 2013*, pages 121–128, 2013.

[BZ05] Yaniv Bernstein and Justin Zobel. Redundant documents and search effectiveness. In *Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management, Bremen, Germany, October 31 - November 5, 2005*, pages 736–743, 2005.

[CBSS14] Chin-Wan Chung, Andrei Z. Broder, Kyuseok Shim, and Torsten Suel, editors. *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014*. ACM, 2014.

[Cen] Pew Research Center. The modern news consumer. http://www.journalism.org/2016/07/07/the-modern-news-consumer/pj_2016-07-07_modern-news-consumer_1-01/.

[CL11] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM TIST*, 2(3):27, 2011.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[DG14] Jesse Dunietz and Daniel Gillick. A new entity salience task with millions of training examples. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*, pages 205–209, 2014.

[DGH$^+$14] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 601–610, 2014.

[DLM13] Sanmay Das, Allen Lavoie, and Malik Magdon-Ismail. Manipulation among the arbiters of collective intelligence: how wikipedia administrators mold public opinion. In *22nd CIKM, San Francisco, USA*, 2013.

[DMBZ10] Gianluca Demartini, Malik Muhammad Saad Missen, Roi Blanco, and Hugo Zaragoza. Entity summarization of news articles. In *Proceeding of the 33rd ACM SIGIR*, pages 795–796, 2010.

[DRSZ13] Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. Recognizing textual entailment: Models and applications. *Synthesis Lectures on Human Language Technologies*, 6(4):1–220, 2013.

[Enc] Wikipedia: The Free Encyclopedia. Wikipedia: The free encyclopedia. https://www.wikipedia.org.

[ESF+08] Aaron Elkiss, Siwei Shen, Anthony Fader, Günes Erkan, David J. States, and Dragomir R. Radev. Blind men and elephants: What do citation summaries tell us about a research article? *JASIST*, 59(1):51–62, 2008.

[FAA15] Besnik Fetahu, Abhijit Anand, and Avishek Anand. How much is wikipedia lagging behind news? In *Proceedings of the ACM Web Science Conference, WebSci 2015, Oxford, United Kingdom, June 28 - July 1, 2015*, pages 28:1–28:9, 2015.

[FGD15] Besnik Fetahu, Ujwal Gadiraju, and Stefan Dietze. Improving entity retrieval on structured data. In *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I*, pages 474–491, 2015.

[FGM05] Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, pages 363–370, 2005.

[FMA15] Besnik Fetahu, Katja Markert, and Avishek Anand. Automated news suggestions for populating wikipedia entity pages. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, pages 323–332, 2015.

[FMNA16] Besnik Fetahu, Katja Markert, Wolfgang Nejdl, and Avishek Anand. Finding news citations for wikipedia. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, pages 337–346, 2016.

[FS12] Paolo Ferragina and Ugo Scaiella. Fast and accurate annotation of short texts with wikipedia pages. *IEEE Software*, 29(1):70–75, 2012.

[FSMM13] Heather Ford, Shilad Sen, David R. Musicant, and Nathaniel Miller. Getting to the source: where does wikipedia get its information from? In *Proceedings of the 9th International Symposium on Open Collaboration, Hong Kong, China, August 05 - 07, 2013*, pages 9:1–9:10, 2013.

[Gar55] Eugene Garfield. Citation indexes for science: A new dimension in documentation through association of ideas. *Science*, 122(3159):108–111, 1955.

[GGSL12] Christophe Guéret, Paul T. Groth, Claus Stadler, and Jens Lehmann. Assessing linked data mappings using network measures. In *Proceedings of the 9th ESWC*, pages 87–102, 2012.

[Har75] Stephen P. Harter. A probabilistic approach to automatic keyword indexing. part i. on the distribution of specialty words in a technical literature. *JASIS*, 26(4):197–206, 1975.

[Har12] Andreas Harth. Billion Triples Challenge data set. Downloaded from http://km.aifb.kit.edu/projects/btc-2012/, 2012.

[HAW14] Johannes Hoffart, Yasemin Altun, and Gerhard Weikum. Discovering emerging entities with ambiguous names. In Chung et al. [CBSS14], pages 385–396.

[HCMB03] Monika Rauch Henzinger, Bay-Wei Chang, Brian Milch, and Sergey Brin. Query-free news search. In *Proceedings of the Twelfth International World Wide Web Conference, WWW 2003, Budapest, Hungary, May 20-24, 2003*, pages 1–10, 2003.

[HHM+10] Harry Halpin, Daniel M. Herzig, Peter Mika, Roi Blanco, Jeffrey Pound, Henry S. Thompson, and Thanh Tran Duc. Evaluating ad-hoc object retrieval. In *Proceedings of the IWEST*, 2010.

[HKC+12] Wenyi Huang, Saurabh Kataria, Cornelia Caragea, Prasenjit Mitra, C. Lee Giles, and Lior Rokach. Recommending citations: translating papers into references. In *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*, pages 1910–1914, 2012.

[HLS+07] Meiqun Hu, Ee-Peng Lim, Aixin Sun, Hady Wirawan Lauw, and Ba-Quy Vuong. Measuring article quality in wikipedia: models and evaluation. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007*, pages 243–252, 2007.

[Hub17] Christoph Hube. Bias in wikipedia. In *Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, April 3-7, 2017*, pages 717–721, 2017.

[HW79] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[HWMG14] Wenyi Huang, Zhaohui Wu, Prasenjit Mitra, and C. Lee Giles. Refseer: A citation recommendation system. In *IEEE/ACM Joint Conference* *on Digital Libraries, JCDL 2014, London, United Kingdom, September 8-12, 2014*, pages 371–374, 2014.

[HYB+11] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 782–792, 2011.

[Kat08] Rohit J. Kate. A dependency-based word subsequence kernel. In *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25-27 October 2008, Honolulu, Hawaii, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 400–409, 2008.

[KGC11] Brian Keegan, Darren Gergle, and Noshir S. Contractor. Hot off the wiki: dynamics, practices, and structures in wikipedia's coverage of the tōhoku catastrophes. In *Proceedings of the 7th International Symposium on Wikis and Open Collaboration, 2011, Mountain View, CA, USA, October 3-5, 2011*, pages 105–113, 2011.

[KGC12] Brian Keegan, Darren Gergle, and Noshir S. Contractor. Do editors or articles drive collaboration?: multilevel statistical network analysis of wikipedia coauthorship. In *CSCW '12 Computer Supported Cooperative Work, Seattle, WA, USA, February 11-15, 2012*, pages 427–436, 2012.

[KMN+02] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):881–892, 2002.

[KPSM07] Aniket Kittur, Bryan A. Pendleton, Bongwon Suh, and Todd Mytkowicz. Power of the few vs. wisdom of the crowd: Wikipedia and the rise of the bourgeoisie. In *SIGCHI*, 2007.

[KTT16] Dain Kaplan, Takenobu Tokunaga, and Simone Teufel. Citation block determination using textual coherence. *JIP*, 24(3):540–553, 2016.

[LMP01] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289, 2001.

[LN89] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

[LRU14] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2014.

[LS] Accessed 20/04/2017 LOD Stats. Linked open data statistics. http://stats.lod2.eu.

[LT10] Brendan Luyt and Daniel Tan. Improving wikipedia's credibility: References and citations in a sample of history articles. *JASIST*, 61(4), 2010.

[MJGB11] Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings the 7th International Conference on Semantic Systems, I-SEMANTICS 2011, Graz, Austria, September 7-9, 2011*, pages 1–8, 2011.

[ML03] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 188–191. Association for Computational Linguistics, 2003.

[MOM$^+$15] Mostafa Mesgari, Chitu Okoli, Mohamad Mehdi, Finn Årup Nielsen, and Arto Lanamäki. "the sum of all human knowledge": A systematic review of scholarly research on the content of wikipedia. *JASIST*, 66(2), 2015.

[MRS08] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[MT04] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain*, pages 404–411, 2004.

[MW08] David N. Milne and Ian H. Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008*, pages 509–518, 2008.

[Ng10] Vincent Ng. Supervised noun phrase coreference research: The first fifteen years. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 1396–1411, 2010.

[NMS$^+$07] Zaiqing Nie, Yunxiao Ma, Shuming Shi, Ji-Rong Wen, and Wei-Ying Ma. Web object retrieval. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, pages 81–90, 2007.

[NO99] Hidetsugu Nanba and Manabu Okumura. Towards multi-paper summarization using reference information. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI 99, Stockholm, Sweden, July 31 - August 6, 1999. 2 Volumes, 1450 pages*, pages 926–931, 1999.

[NPM07] Ani Nenkova, Rebecca J. Passonneau, and Kathleen McKeown. The pyramid method: Incorporating human content selection variation in summarization evaluation. *TSLP*, 4(2):4, 2007.

[NW08] Thomas Neumann and Gerhard Weikum. Rdf-3x: A risc-style engine for rdf. *Proc. VLDB Endow.*, 1(1):647–659, August 2008.

[nyt] New york times annotated corpus. *http://corpus.nytimes.com*.

[O'C82] John O'Connor. Citing statements: Computer recognition and use to improve retrieval. *Inf. Process. Manage.*, 18(3):125–131, 1982.

[Oka07] Naoaki Okazaki. Crfsuite: a fast implementation of conditional random fields (crfs). 2007.

[oW] List of Wikipedias. List of wikipedias. https://en.wikipedia.org/wiki/List_of_Wikipedias.

[PBMW99] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. 1999.

[PDL$^+$08] Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K. Joshi, and Bonnie L. Webber. The penn discourse treebank 2.0. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco*, 2008.

[PHT09] Katherine A. Panciera, Aaron Halfaker, and Loren G. Terveen. Wikipedians are born, not made: a study of power editors on wikipedia. In *Proceedings of the 2009 International ACM SIGGROUP Conference on Supporting Group Work, GROUP 2009, Sanibel Island, Florida, USA, May 10-13, 2009*, pages 51–60, 2009.

[PM$^+$00] Dan Pelleg, Andrew W Moore, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML*, pages 727–734, 2000.

[PMZ10] Jeffrey Pound, Peter Mika, and Hugo Zaragoza. Ad-hoc object retrieval in the web of data. In *Proceedings of the 19th WWW*, pages 771–780, 2010.

[PN09] Emily Pitler and Ani Nenkova. Using syntax to disambiguate explicit discourse connectives in text. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore, Short Papers*, pages 13–16, 2009.

[POK$^+$13] Silvia Pareti, Timothy O'Keefe, Ioannis Konstas, James R. Curran, and Irena Koprinska. Automatically detecting and attributing indirect quotations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 989–999, 2013.

[pol] Wikipedia:Editing policy. Wikipedia:editing policy. https://en.wikipedia.org/wiki/Wikipedia:Editing_policy.

[Pro] The Million Books Project. The million books project. http://archive.org/details/millionbooks.

[PW11] Philipp Petrenz and Bonnie Webber. Stable classification of text genres. *Computational Linguistics*, 37(2):385–393, 2011.

[QR08] Vahed Qazvinian and Dragomir R. Radev. Scientific paper summarization using citation summary networks. In *COLING 2008, 22nd International Conference on Computational Linguistics, Proceedings of the Conference, 18-22 August 2008, Manchester, UK*, pages 689–696, 2008.

[QR10] Vahed Qazvinian and Dragomir R. Radev. Identifying non-explicit citing sentences for citation-based summarization. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 555–564, 2010.

[Rib14] Bruno Ribeiro. Modeling and predicting the growth and death of membership-based websites. In Chung et al. [CBSS14], pages 653–664.

[RRT08] Anna Ritchie, Stephen Robertson, and Simone Teufel. Comparing citation contexts for information retrieval. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008*, pages 213–222, 2008.

[RTR06] Anna Ritchie, Simone Teufel, and Stephen Robertson. How to find better index terms through citations. In *Proceedings of the Workshop on How Can Computational Linguistics Improve Information Retrieval?*, CLIIR '06, pages 25–32, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[RU11] Anand Rajaraman and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2011.

[RWJ+95] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. Okapi at trec-3. *Nist Special Publication Sp*, 109:109, 1995.

[RZ09] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, April 2009.

[San08] Evan Sandhaus. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752, 2008.

[SB09] Christina Sauper and Regina Barzilay. Automatically generating wikipedia articles: A structure-aware approach. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*, pages 208–216, 2009.

[SCCP09] Bongwon Suh, Gregorio Convertino, Ed H. Chi, and Peter Pirolli. The singularity is not near: Slowing growth of wikipedia. In *Proceedings of the 5th International Symposium on Wikis and Open Collaboration*, WikiSym '09, pages 8:1–8:10, New York, NY, USA, 2009. ACM.

[SG10] Jannik Strötgen and Michael Gertz. Heideltime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010, Uppsala University, Uppsala, Sweden, July 15-16, 2010*, pages 321–324, 2010.

[Sim71] Herbert A Simon. Designing organizations for an information-rich world. 1971.

[Sit] Alexa Internet Ranking Site. Alexa topsites. accessed at 12.05.2017 from http://www.alexa.com/topsites.

[SKW07] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th International*

Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007, pages 697–706, 2007.

[Sta] Internet World Stats. Internet world stats. http://internetworldstats.com/stats.htm.

[SWM10] Serge Sharoff, Zhili Wu, and Katja Markert. The web library of babel: evaluating genre collections. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta*, 2010.

[SWY75] Gerard Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975.

[TDCM12] Alberto Tonon, Gianluca Demartini, and Philippe Cudré-Mauroux. Combining inverted indices and structured search for ad-hoc object retrieval. In *Proceedings of the 35th ACM SIGIR*, pages 125–134, 2012.

[TKMS03] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2003, Edmonton, Canada, May 27 - June 1, 2003*, 2003.

[TM00] Kristina Toutanova and Christopher D Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*, pages 63–70. Association for Computational Linguistics, 2000.

[TW13] Bilyana Taneva and Gerhard Weikum. Gem-based entity-knowledge maintenance. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, pages 149–158, 2013.

[VBZ$^+$15] Michela Del Vicario, Alessandro Bessi, Fabiana Zollo, Fabio Petroni, Antonio Scala, Guido Caldarelli, H. Eugene Stanley, and Walter Quattrociocchi. Echo chambers in the age of misinformation. *CoRR*, abs/1509.00189, 2015.

[vL07] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[W3Ca] W3C. Rdf 1.1. concepts and abstract syntax. http://www.w3.org/TR/rdf11-concepts/.

[W3Cb] W3C. Rdf schema 1.1. https://www.w3.org/TR/rdf-schema/.

[WD08] Pu Wang and Carlotta Domeniconi. Building semantic kernels for text classification using wikipedia. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, pages 713–721, 2008.

[WGM+14] Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. Knowledge base completion via search-based question answering. In Chung et al. [CBSS14], pages 515–526.

[Wik] Wikipedia. Wikipedia statistics, accessed 12-04-2017. https://en.wikipedia.org/wiki/Wikipedia:Statistics.

[WJP98] Marilyn A Walker, Aravind Krishna Joshi, and Ellen Friedman Prince. *Centering theory in discourse*. Oxford University Press, 1998.

[WM08] Ian Witten and David Milne. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA*, pages 25–30, 2008.

[WRTH15] Morten Warncke-Wang, Vivek Ranjan, Loren G. Terveen, and Brent Hecht. Misalignment between supply and demand of quality content in peer production communities. In *Proceedings of the Ninth International Conference on Web and Social Media, ICWSM 2015, University of Oxford, Oxford, UK, May 26-29, 2015*, pages 493–502, 2015.

[YP97] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML 1997), Nashville, Tennessee, USA, July 8-12, 1997*, pages 412–420, 1997.

[ZA13] Nikita Zhiltsov and Eugene Agichtein. Improving entity search over linked data by modeling latent semantics. In *Proceedings of the 22nd ACM CIKM*, pages 1253–1256, 2013.

[ZH06] Liang Zhou and Eduard H Hovy. On the summarization of dynamically introduced information: Online discussions and blogs. In *AAAI Spring symposium: Computational approaches to analyzing weblogs*, page 237, 2006.

[ZPL12] Dell Zhang, Karl Prior, and Mark Levene. How long do wikipedia editors keep active? In *Proceedings of the Eighth Annual International Symposium on Wikis and Open Collaboration, WikiSym 2012, Austria, August 27 - 29, 2012*, page 4, 2012.