UNIVERSITÀ DEGLI STUDI
DI TRENTO

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE
ICT International Doctoral School

Automatic Population of Structured Knowledge Bases via Natural Language Processing

Marco Fossati

Advisor

Giovanni Tummarello

Fondazione Bruno Kessler

Abstract

The Web has evolved into a huge mine of knowledge carved in different forms, the predominant one still being the free-text document. This motivates the need for Intelligent Web-reading Agents: hypothetically, they would skim through disparate Web sources corpora and generate meaningful structured assertions to fuel Knowledge Bases (KBs). Ultimately, comprehensive KBs, like Wikidata and DBpedia, play a fundamental role to cope with the issue of information overload. On account of such vision, this thesis depicts a set of systems based on Natural Language Processing (NLP), which take as input unstructured or semi-structured information sources and produce machine-readable statements for a target KB. We implement four main research contributions: (1) a one-step methodology for crowdsourcing the Frame Semantics annotation; (2) a NLP technique implementing the above contribution to perform N-ary Relation Extraction from Wikipedia, thus enriching the target KB with properties; (3) a taxonomy learning strategy to produce an intuitive and exhaustive class hierarchy from the Wikipedia category graph, thus augmenting the target KB with classes; (4) a recommender system that leverages a KB network to yield atypical suggestions with detailed explanations, serving as a proof of work for real-world end users. The outcomes are incorporated into the Italian DBpedia chapter, can be queried through its public endpoint, and/or downloaded as standalone data dumps.

Keywords

Natural Language Processing, Information Extraction, Machine Learning, Frame Semantics, Crowdsourcing, Recommender Systems, Wikipedia.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

The World Wide Web (WWW) is nowadays one of the most prominent sources of information and knowledge. Since its birth, the amount of publicly available data has dramatically increased and has led to the problem of information overload. Users are no longer able to handle such a huge volume of data and need to spend time finding the right piece of information which is relevant to their interests. Furthermore, a major portion of the WWW content is represented as unstructured data, namely free-text documents, together with multimedial data such as images, audio and video. Understanding its meaning is a complex task for machines and still relies on subjective human interpretations. The *Web of Data* envisions its evolution as a repository of machine-readable structured data. This would enable an automatic and unambiguous content analysis and its direct delivery to end users.

The idea has not only engaged a long strand of research, but has also been absorbed by the biggest web industry players. Companies such as Google, Facebook and Microsoft, have already adopted large-scale semantics-driven systems, namely Google's Knowledge Graph,[1] Facebook's Graph

[1] https://www.google.com/intl/en_us/insidesearch/features/search/knowledge.html

SEARCH,[2] and Microsoft's SATORI.[3] Moreover, the WWW Consortium, together with the Linked Data[4] (LD) initiative, has provided a standardized technology stack to publish freely accessible interconnected datasets. LD is becoming an increasingly popular paradigm to disseminate Open Data (OD) produced by all kinds of public organizations.

The international Linked Open Data (LOD) Cloud[5] counts today several billion records from hundreds of sources. It is worth to note that the phenomenon is not limited to public organizations: over recent years, a number of game-changing announcements has been broadcast by private companies, thus potentially contributing to augment the Linked Data ecosystem. First, Google's Knowledge Graph stems from the acquisition of one of the most important nodes of the LOD cloud, namely FREEBASE;[6] secondly, the coalition between the largest search engines Google, Bing, Yahoo! and Yandex, has led to the introduction of SCHEMA.ORG,[7] a combination of a vocabulary and a set of incentives for web publishers to annotate their content with metadata markup; finally, large private organizations are approaching LD, by evolving their business models or by modifying their production processes to comply with the openness of the LOD cloud.

In this scenario, a *Knowledge Base* (KB) is a repository that encodes areas of human intelligence in a graph structure, where real-world and abstract entities are bound together through relationships, and classified according to a formal description of the world, i.e., an *ontology*. KBs bear a considerable impact in everyday's life, since they power a steadily growing

[2] http://www.facebook.com/about/graphsearch

[3] https://blogs.bing.com/search/2015/08/20/bing-announces-availability-of-the-knowledge-and-action-graph-api-for-developers/

[4] http://linkeddata.org

[5] http://lod-cloud.net/

[6] http://www.freebase.com/

[7] http://schema.org/

number of applications, from Web search engines to question answering platforms, all the way to digital library archives and data visualization facilities, just to name a few. Under this perspective, Wikipedia is the result of a crowdsourced effort and stands for the best digital materialization of encyclopedic human knowledge. Therefore, the general-purpose nature of its content plays a vital role for powering a KB. Furthermore, it is released under the Creative Commons BY-SA license,[8] that enables free reuse and redistribution. Hence, it is not surprising that its data has been attracting both research and industry interests, and has driven the creation of several KBs, the most prominent being BabelNet [87], DBpedia [73], Freebase [14], YAGO [63], Wikidata [125], and WikiNet [85], among others.

In particular, the main contribution of DBpedia[9] is to automatically extract structured data from semi-structured Wikipedia content, typically *infoboxes*.[10] DBpedia acts as the central component of the growing LOD cloud and benefits from a steadily increasing multilingual community of users and developers. Its stakeholders range from journalists [55] to governmental institutions [38], up to digital libraries [57]. Its international version was first conceived as a multilingual resource, assembling information coming from diverse Wikipedia localizations. On one hand, multilingualism would naturally be of universal impact to the society, as it can nurture users at a worldwide scale. On the other hand, it does not only represent an enormous cultural challenge, but also a technological one, as it would require to merge radically different views of the world into one single classification schema (i.e., the ontology). As such, the focal point of DBpedia was initially set to the English chapter, since it is the richest one with respect to the total number of articles. Therefore, multilingual data was restricted to those

[8] http://creativecommons.org/licenses/by-sa/3.0/legalcode
[9] http://dbpedia.org
[10] http://en.wikipedia.org/wiki/Help:Infobox

items that have a counterpart (i.e., an interlanguage link) in the English branch. Recently, an internationalization effort [70] has been conducted to tackle the problem and has led to the birth of several language-specific deployments: among them, the author of this thesis has developed and maintains the Italian DBpedia chapter.[11]

Besides the interest of the KB itself, the *Italian DBpedia* is a publicly available resource of critical importance for the national LD initiative. Thanks to its encyclopedic cross-domain nature, it may serve as a hub to which other datasets can link, following the same fashion as the international chapter. Consequently, this would cater for the integration of freely accessible data coming from third-party sources in order to ensure textual content augmentation. In addition, the Italian Wikipedia is the seventh most impactful chapter worldwide in terms of content (if we exclude automatically built ones), with more than 1,23 million articles,[12] and the eight with respect to usage.[13]

1.1 The Vision

This thesis acts as a seed that would burgeon as a country-centric KB with large amounts of real-world entities of national and local interest. The KB would empower a broad spectrum of applications, from data-driven journalism to public library archives enhancement, not to forget data visualization amenities. The language-specific Wikipedia chapter will serve as its core. Such resource will allow the deployment of a central data hub acting as a reference access point for the user community. Moreover, it will foster the amalgamation of publicly available external resources, resulting in a rich content enhancement. Governmental and research OD

[11] http://it.dbpedia.org
[12] https://meta.wikimedia.org/wiki/List_of_Wikipedias
[13] http://stats.wikimedia.org/EN/Sitemap.htm

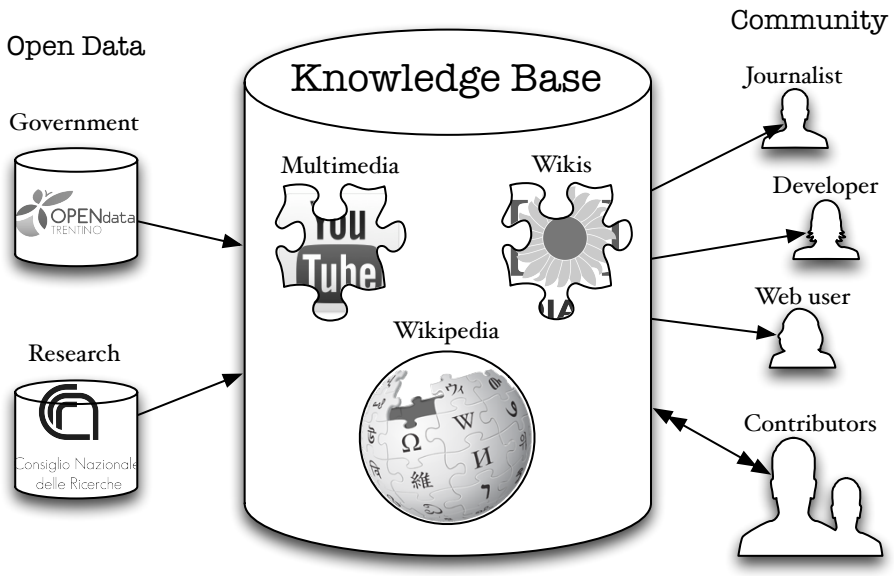

Figure 1.1: Thesis vision

will be interlinked to the KB as well. Ultimately, data consumers such as journalists, digital libraries, software developers or web users in general will be able to leverage the KB as input for writing articles, enriching a catalogue, building applications or simply satisfying their information retrieval needs. Figure 1.1 depicts this vision.

Given the above premises, we foresee the following main outcomes as starting points for further development:

1. deployment of a high-quality Italian DBpedia acting as the backbone of a healthy LOD environment. The Italian case will then serve as a best practice for full internationalization;

2. completely data-driven approaches for KB enrichment. More specifically:

   - a linguistically-oriented relation extraction methodology for property population;
   - a taxonomy learning strategy for classes population.

3. Liaison with civic society organizations in order to make the KB the gold-standard nucleus for the interlinkage of national OD initiatives.

1.2 The Problem

The de facto model underpinning the classification of all the multilingual encyclopedic entries, namely the DBpedia ontology (DBPO),[14] **is exceedingly unbalanced**. This is attributable to the collaborative nature of its development and maintenance: any registered contributor can edit it by adding, deleting or modifying its content. At the time of writing this thesis (July 2016), the latest DBPO release[15] contains 739 classes and 2,827 properties, which are highly heterogeneous in terms of granularity (cf. for instance the classes BAND versus SAMBASCHOOL, both direct subclasses of ORGANISATION) and are supposed to encapsulate the entire encyclopedic world. This indicates there is ample room to improve the quality of DBPO.

Furthermore, a clear problem of **class and property coverage** has been recently pointed out [94, 5, 99, 51]: each Wikipedia *entry* should have a 1-to-1 mapping to each DBpedia *entity*. However, this is not reflected in the current state of affairs: for instance, although the English Wikipedia contains more than 5 million articles, DBpedia has only classified 2.8 million into DBPO. One of the major reasons is that a significant amount of Wikipedia entries does not contain an infobox, which is a valuable piece of information to infer a meaningful description of an entry. This results in a large number of DBpedia entities with poor or no data, thus restraining the exploitation of the KB, as well as limiting its usability potential. The current classification paradigm described in [73] heavily depends on Wikipedia infobox names and attributes in order to enable a manual mapping to DBPO classes and properties. The availability and

[14] http://mappings.dbpedia.org/server/ontology/classes/
[15] http://wiki.dbpedia.org/dbpedia-dataset-version-2015-10

homogeneity of such semi-structured data in Wikipedia pages is unstable for two reasons, namely (a) the community-based, manually curated nature of the project, and (b) the linguistic and cultural discrepancies among all the language chapters. This triggers several shortcomings, as highlighted in [51]. Furthermore, resources can be wrongly classified or defectively described, as a result of (a) the misuse of infoboxes by Wikipedia contributors, (b) overlaps among the four mostly populated DBPO classes, namely PLACE, PERSON, ORGANISATION and WORK,[16] and, most importantly, (c) the lack of suitable mappings from Wikipedia infoboxes to DBPO. Consequently, the extension of the DBpedia data coverage is a crucial step towards the release of richly structured and high quality data.

From a socio-political perspective, the Italian initiative is flourishing in the global OD landscape, with 15,000 datasets notified by public administrations,[17] not counting other initiatives from organizations such as digital libraries. However, from a technical outlook, the vast majority of such data is extracted from databases and made available in flat tabular formats (e.g., CSV), which are not always adequate to fully express the complex structure and semantics of the original data. The star model[18] foresees the publication of OD according to a 5-level quality scale: (1) use an open license, (2) expose semi-structured tabular data, (3) use non-proprietary formats, (4) mint URIs for data representation, and (5) connect to other datasets that are already exposed as LOD. This deployment model suggests to go beyond tabular data (3 stars) and to adopt the principles of LD.

Various national public administrations have already started to publish their 5-star OD: for instance, the recent efforts of the province of Trento[19]

[16] http://wiki.dbpedia.org/Datasets39/DatasetStatistics

[17] https://joinup.ec.europa.eu/sites/default/files/ckeditor_files/files/eGov%20in%20Italy%20-%20April%202015%20-%20v_17_1.pdf

[18] http://5stardata.info

[19] http://dati.trentino.it/

and the municipality of Florence[20] are among the most mature examples of governmental OD in Italy. Interestingly, all of them are already linking their dataset to DBpedia. As this phenomenon continues to grow, there is an ever growing need for a central hub which can be used to disambiguate entity references. We believe that the encyclopedic general-purpose nature of the Italian DBpedia makes it the ideal candidate for becoming a national semantic entity hub, very much like the international DBpedia project naturally became the nucleus for the international LOD Cloud.

Likewise, the Italian DBpedia would meet the needs of the *Digital Agenda for Europe* initiative, which argues in action 26[21] that member states should align their national interoperability frameworks to the European one (EIF). The National Interoperability Framework Observatory has recently analyzed the Italian case,[22] highlighting a weak alignment to EIF with regards to interoperability levels. On that account, the national governmental institution *Agency for Digital Italy*[23] has published a set of guidelines concentrating on semantic interoperability.[24]

1.3 The Solution and its Innovative Aspects

We investigate the problems of DBPO **heterogeneity** and **lack of coverage** by means of a practical outcome, namely the **deployment of a high-quality structured KB extracted from the Italian Wikipedia**. This has been carried out under the umbrella of the DBpedia open source

[20] http://opendata.comune.fi.it/

[21] http://ec.europa.eu/digital-agenda/en/pillar-ii-interoperability-standards/action-26-ms-implement-european-interoperability-framework

[22] https://joinup.ec.europa.eu/sites/default/files/3b/66/1d/NIFO%20-%20Factsheet%20Italy%2005-2013.pdf

[23] http://www.agid.gov.it/

[24] http://www.agid.gov.it/sites/default/files/documentazione_trasparenza/cdc-spc-gdl6-interoperabilitasemopendata_v2.0_0.pdf

organization.[25] The author has founded and maintains the *Italian DBpedia chapter*[26] and is member of the DBpedia Association board of trustees.[27]

1.3.1 Contributions

The outcomes that constitute the main research contributions of this dissertation and have brought the Italian DBpedia resource to its current status are broken down as follows.

Contribution 1: a one-step methodology for *crowdsourcing* a complex linguistic task to the layman, namely full *Frame Semantics* annotation;

Contribution 2: a *NLP* technique implementing the above methodology to automatically perform *N-ary Relation Extraction* from free text, applied to enrich the KB with *properties*;

Contribution 3: a *Taxonomy Learning* strategy to automatically generate and populate an intuitive wide-coverage class hierarchy from the Wikipedia category[28] graph, applied to enrich the KB with *classes*;

Contribution 4: a novel *Recommender System* that leverages an external KB to provide unusual recommendations and exhaustive explanations: while the implemented use case does not directly exploit the Italian DBpedia, this contribution represents a direct application of our main efforts and demonstrates the potential for real-world end users.

It should be highlighted that part of this thesis has already been assessed by the scientific community via standard peer-review procedures. We list below the publications and connect them to the aforementioned contributions in Table 1.1:

[25] http://dbpedia.org

[26] http://it.dbpedia.org

[27] https://docs.google.com/document/d/1pchRPLtQwO3GH49cF7GB33srRn1p2M3QW8Jtu5b5ZwE/edit

[28] https://en.wikipedia.org/wiki/Help:Categories

1. Marco Fossati, Claudio Giuliano, and Sara Tonelli. Outsourcing Framenet to the Crowd. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 2013 (acceptance rate: 24%);

2. Marco Fossati, Sara Tonelli, and Claudio Giuliano. Frame Semantics Annotation Made Easy With DBpedia. In *Proceedings of the 1st International Workshop on Crowdsourcing the Semantic Web* at ISWC, 2013;

3. Marco Fossati, Emilio Dorigatti, and Claudio Giuliano. N-ary Relation Extraction for Simultaneous T-Box and A-Box Knowledge Base Augmentation. *Semantic Web Journal* (under review)

4. Marco Fossati, Dimitris Kontokostas, and Jens Lehmann. Unsupervised Learning of an Extensive and Usable Taxonomy for DBpedia. In *Proceedings of the 11th International Conference on Semantic Systems*, 2015. (Acceptance rate: 26%) **Best paper nominee**;

5. Marco Fossati, Claudio Giuliano, and Giovanni Tummarello. Semantic Network-driven News Recommender Systems: a Celebrity Gossip Use Case. In *Proceedings of the International Workshop on Semantic Technologies meet Recommender Systems & Big Data* at ISWC, 2012.

Table 1.1: Research contributions and associated publications

Contribution	**Publications**
Crowdsourced frame annotation	#1, #2
Properties population via Relation Extraction	#3
Classes population via Taxonomy Learning	#4
Application to Recommender Systems	#5

StrepHit: Contribution 2 Got Funded by the Wikimedia Foundation It is worth to pinpoint that we won the **largest Wikimedia Foundation Individual Engagement Grant**, 2015 round 2 call, to pursue our research based on Contribution 2, under the umbrella of Wikidata. The selected project proposal stems from the lessons learnt in article #3 and aims at developing a Web-reading agent to corroborate Wikidata content with external references. Full details are available in Chapter 8.

1.4 Structure of the Thesis

The remainder of this work is structured as follows.

Chapter 2 provides an overview of related efforts, spanning over the different research areas. The reader may then find more details in each specific chapter;

Chapter 3 illustrates the crowdsourcing methodology to perform a complete annotation of frame semantics in natural language utterances. This chapter coincides to papers #1 and #2;

Chapter 4 describes the NLP pipeline that aims at populating DBpedia with properties. It implements the above crowdsourcing methodology and achieves N-ary Relation Extraction given a Wikipedia free text corpus. This chapters embeds article #3;

Chapter 5 contains the taxonomy learning system that enriches DBpedia with classes. It processes the Wikipedia category graph, generates a class hierarchy and populates it with instance assertions. This chapter corresponds to paper #4;

Chapter 6 outlines an end-user application that leverages a target KB to deliver news articles recommendations, coupled with informative

explanations. This chapter encompasses paper #5;

Chapter 7 sums up the results of this thesis and points out specific open issues to be further developed;

Chapter 8 embeds the StrepHit technical reports.

Chapter 2

State of the Art

Due to its interdisciplinary nature, our work embraces different research areas, which however are strictly interconnected. The *fil rouge* that binds them is Natural Language Processing (NLP), i.e., a set of practices allowing machines to understand human language. More specifically, most of our contributions leverage off-the-shelf Entity Linking (EL) techniques. In this chapter, we provide a high-level overview of the technical background, with pointers to the most relevant related work. The reader may then dive into the specific ones for more detailed comparison.

2.1 Entity Linking

EL is the task of matching free-text chunks to entities of a target KB. This is formulated as a word sense disambiguation (WSD) problem: the meaning of an input set of words (i.e., an n-gram) is resolved through an unambiguous link to the KB. Several efforts have adopted Wikipedia to build WSD systems, with seminal work in [30, 15]. It should be mentioned that linking to Wikipedia implies linking to DBpedia, as the only difference relies in a part of the URI (i.e., `wikipedia.org/wiki` versus `dbpedia.org/resource`). A considerable amount of full EL tools have stemmed from both research and

industry, such as BABELFY[1] [84], DBPEDIA SPOTLIGHT[2] [31, 81], THE WIKI MACHINE[3] [54] and ALCHEMY[4], COGITO,[5] DANDELION,[6] OPEN CALAIS[7] respectively. A comparative performance evaluation is detailed in [82].

2.2 Frame Semantics

Frame semantics [46] is one of the theories that originates from the long strand of linguistic research in artificial intelligence. A *frame* can be informally defined as an event triggered by some term in a text and embedding a set of participants. For instance, the sentence `Goofy has murdered Mickey Mouse` evokes the KILLING frame (triggered by `murdered`) together with the Killer and Victim participants (respectively `Goofy` and `Mickey Mouse`). Such theory has led to the creation of FrameNet [8], namely an English lexical database containing manually annotated textual examples of frame usage.

Currently, FrameNet development follows a strict protocol for data annotation and quality control. The entire procedure is known to be both time-consuming and costly, thus representing a burden for the extension of the resource [7]. Furthermore, deep linguistic knowledge is needed to tackle this annotation task, and the resource developed so far would not have come to light without the contribution of skilled linguists and lexicographers. On one hand, the task complexity depends on the inherently complex theory behind frame semantics, with a repository of thousands of roles available for the assignment. On the other hand, these roles are defined for expert

[1] http://babelfy.org/
[2] http://spotlight.dbpedia.org/
[3] http://thewikimachine.fbk.eu/
[4] http://www.alchemyapi.com/
[5] http://www.cogitoapi.com/
[6] https://dandelion.eu/
[7] http://www.opencalais.com/

annotators, and their descriptions are often obscure to common readers. We report three examples below:

- Support: Support is a fact that lends epistemic support to a claim, or that provides a reason for a course of action. Typically it is expressed as an External Argument. (EVIDENCE frame)

- Protagonist: A person or self-directed entity whose actions may potentially change the mind of the Cognizer (INFLUENCE_OF_EVENT_ON_COGNIZER frame)

- Locale: A stable bounded area. It is typically the designation of the nouns of Locale-derived frames. (LOCALE_BY_USE frame)

2.3 Crowdsourcing

In computer science, the term *crowdsourcing* encodes all the activities which are difficult for machines to be solved, but easier for humans, and are cast to a non-specialized crowd.

The construction of annotation datasets for NLP tasks via non-expert contributors has been approached in different ways, the most prominent being games with a purpose (GWAP) and micro-tasks. While the former technique leverages fun as the motivation for attracting participants, the latter mainly relies on a monetary reward. The effects of such factors on a contributor's behavior have been analyzed in the motivation theory literature, but are beyond the scope of this thesis. The reader may refer to [68] for an overview focusing on a specific platform, namely AMAZON'S MECHANICAL TURK.[8]

[8] https://www.mturk.com/mturk/welcome

2.3.1 Games with a Purpose

Verbosity [124] was one of the first attempts in gathering annotations with a GWAP. Phrase Detectives [24, 23] was meant to harvest a corpus with co-reference resolution annotations. The game included a validation mode, where participants could assess the quality of previous contributions. A data unit, namely a resolved coreference for a given entity, is judged complete only if the agreement is unanimous. Disagreement between experts and the crowd appeared to be a potential indicator of ambiguous input data. Indeed, it has been shown that in most cases disagreement did not represent a poor annotation, but rather a valid alternative.

2.3.2 Micro-tasks

[116] described design and evaluation guidelines for five natural language micro-tasks. Similarly to our approach, the authors compared crowdsourced annotations with expert ones for quality estimation. Moreover, they used the collected annotations as training sets for machine learning classifiers and measured their performance. However, they explicitly chose a set of tasks that could be easily understood by non-expert contributors, thus leaving the recruitment and training issues open. [88] built a multilingual textual entailment dataset for statistical machine translation systems.

2.3.3 Frame Semantics Annotation

The more specific Frame Semantics annotation problem has been recently addressed via crowdsourcing by [64]. Furthermore, [7] highlighted the crucial role of recruiting people from the crowd in order to bypass the need for linguistics expert annotations. Uniformly to our contribution, the task described in [64] was modeled in a multiple-choice answers fashion. Nevertheless, the focus is narrowed to the frame discrimination task,

namely selecting the correct frame evoked by a given lemma. Such task is comparable to the word sense disambiguation one as per [116], although the difficulty seems augmented, due to lower inter-annotator agreement values. We believe the frame elements recognition we are attempting to achieve is a more straightforward solution, thus yielding better results. The authors experienced issues that are related to our work with respect to the quality check mechanism in the CrowdFlower platform,[9] as well as the complexity of the frame names and definitions. Outsourcing the task to the CrowdFlower platform has two major drawbacks, namely the proprietary nature of the aggregated inter-agreement annotation value provided in the response data and the need to manually simplify frame elements definitions that generated low inter-annotation agreement. We aim at applying standard measures such as Cohen's κ [28].

2.4 Information Extraction for KB Population

We locate the contribution detailed in Chapter 4 at the intersection of the following research areas:

- Information Extraction;
- Knowledge Base Construction;
- Open Information Semantification, also known as Open Knowledge Extraction.

2.4.1 Information Extraction

Although the borders are blurred, nowadays we can distinguish two principal Information Extraction paradigms that focus on the discovery of relations

[9] https://crowdflower.com

holding between entities: Relation Extraction (RE) and Open Information Extraction (OIE). While they both share the same purpose, their difference relies in the relations set size, either fixed or potentially infinite. It is commonly argued that the main OIE drawback is the generation of noisy data [39, 126], while RE is usually more accurate, but requires expensive supervision in terms of language resources [3, 119, 126].

Relation Extraction

RE traditionally takes as input a finite set R of relations and a document d, and induces assertions in the form $rel(subj, obj)$, where rel represent binary relations between a subject entity $subj$ and an object entity obj mentioned in d. Hence, it may be viewed as a closed-domain paradigm. Recent efforts [6, 3, 119] have focused on alleviating the cost of full supervision via distant supervision. Distant supervision leverages available KBs to automatically annotate training data in the input documents. This is in contrast to our work, since we aim at enriching the target KB with external data, rather than using it as a source. Furthermore, our relatively cheap crowdsourcing technique serves as a substitute to distant supervision, while ensuring full supervision. Other approaches such as [11, 127] instead leverage text that is not covered by the target KB, like we do.

Open Information Extraction

OIE is defined as a function $f(d)$ over a document d, yielding a set of triples (np_1, rel, np_2), where nps are noun phrases and rel is a relation between them. Known complete systems include OLLIE [79], REVERB [43], and NELL [22]. Recently, it has been discussed that cross-utterance processing can improve the performance through logical entailments [2]. This paradigm is called "open" since it is not constrained by any schemata, but rather attempts to learn them from unstructured data. In addition, it takes as

input heterogeneous sources of information, typically from the Web.

In general, most efforts have focused on English, due to the high availability of language resources. Approaches such as [44] explore multilingual directions, by leveraging English as a source and applying statistical machine translation (SMT) for scaling up to target languages. Although the authors claim that their system does not directly depend on language resources, we argue that SMT still heavily relies on them. Furthermore, all the above efforts concentrate on binary relations, while we generate n-ary ones: under this perspective, EXEMPLAR [35] is a rule-based system which is closely related to ours.

2.4.2 Knowledge Base Construction

DBPEDIA [73], FREEBASE [14] and YAGO [63] represent the most mature approaches for automatically building KBs from Wikipedia. Despite its crowdsourced nature (i.e., mostly manual), WIKIDATA [125] benefits from a rapidly growing community of active users, who have developed several robots for automatic imports of Wikipedia and third-party data. The KNOWLEDGE VAULT [39] is an example of KB construction combining Web-scale textual corpora, as well as additional semi-structured Web data such as HTML tables. Although our system may potentially create a KB from scratch from an input corpus, we prefer to improve the quality of existing resources and integrate into them, rather than developing a standalone one.

Under a different perspective, [90] builds on [29] and illustrate a general-purpose methodology to translate FrameNet into a fully compliant Linked Open Data KB via the SEMION tool [91]. The scope of such work diverges from ours, since we do not target a complete conversion of the frame repository we leverage. On the other hand, we share some transformation patterns in the dataset generation step (Section 4.8), namely we both link

FEs to their frame by means of RDF predicates.

Likewise, FrameBase [109, 108] is a data integration effort, proposing a single model based on Frame Semantics to assemble heterogenous KB schemata. This would overcome the knowledge soup issue [52], i.e., the blend of disparate ways in which structured datasets are published. Similarly to us, it utilizes Neo-Davidsonian representations to encode n-ary relations in RDF. Further options are reviewed but discarded by the authors, including singleton properties [89] and schema.org roles.[10] In contrast to our work, FrameBase also provides automatic facilities which bring back the n-ary relations to binary ones for easier queries. The key purpose is to amalgamate different datasets in a unified fashion, thus essentially differing from our KB augmentation objective.

2.4.3 Open Information Semantification

OIE output can indeed be considered structured data compared to free text, but it still lacks of a disambiguation facility: extracted facts generally do not employ unique identifiers (i.e., URIs), thus suffering from intrinsic natural language polysemy (e.g., `Jaguar` may correspond to the animal or a known car brand).

To tackle the issue, [40] propose a framework that clusters OIE facts and maps them to elements of a target KB. Similarly to us, they leverage EL techniques for disambiguation and choose DBpedia as the target KB. Nevertheless, the authors focus on A-Box population, while we also cater for the T-Box part. Moreover, OIE systems are used as a black boxes, in contrast to our full implementation of the extraction pipeline. Finally, relations are still binary, instead of our n-ary ones.

The main intuition behind Legalo [106, 104] resides in the exploitation of hyperlinks, serving as pragmatic traces of relations between entities, which

[10] https://www.w3.org/wiki/WebSchemas/RolesPattern

are finally induced via NLP. The first version [104] focuses on Wikipedia articles, like we do. In addition, it leverages page links that are manually curated by editors, while we consume Entity Linking output. Ultimately, its property matcher module can be leveraged for KB enrichment purposes. Most recently, a new release [106] expands the approach by (a) taking into account hyperlinks from Entity Linking tools, and (b) handling generic free text input. On account of such features, both Legalo and the Fact Extractor are proceeding towards closely related directions. This paves the way to a novel paradigm called *Open Knowledge Extraction* by the authors, which is naturally bound to the Open Information Semantification one introduced in [40]. The only difference again relies on the binary nature of Legalo's extracted relations, which are generated upon FRED [53, 105].

FRED is a machine reader that harnesses several NLP techniques to produce RDF graphs out of free text input. It is conceived as a domain-independent middleware enabling the implementation of specific applications. As such, its scope diverges from ours: we instead deliver datasets that are directly integrated into a target KB. In a fashion similar to our work, it encodes knowledge based on Frame Semantics and employs Entity Linking to mint unambiguous URIs for entities and properties. Furthermore, it relies on the same design pattern for expressing n-ary relations in RDF [62]. As opposed to us, it also encodes NLP tools output via standard formats, i.e., EARMARK [97] and NIF [60]. Additionally, it uses a different natural language representation (i.e., Discourse Representation Structures), which requires a deeper layer of NLP technology, namely syntactic parsing, while we stop to shallow processing via grammatical analysis.

Frame Semantics Classification

Supervised machine learning[11] algorithms have been consistently exploited for automatic frame and FEs classification. While they may mitigate the manual annotation effort by reason of their automatic nature, they yet require pre-annotated training data. In addition, state-of-the-art systems still suffer from performance issues. Recently, an approach proposed in [27] encapsulates frame semantics at the whole discourse level. However, it reached a relatively low precision value, namely .41 in an optimal evaluation scenario. In the SemEval-2010 event identification task [110], the system that performed best achieved a precision of .65. Its most recent implementation [33] gained a slight improvement (.70), but we argue that it is still not sufficiently accurate to substitute manual annotation.

2.4.4 Further Approaches

Distributional Methods

An additional strand of research encompasses distributional methods: these originate from Lexical Semantics and can be put to use for Information Extraction tasks. Prominent efforts, e.g., [1, 9, 96] aim at processing corpora to infer features for terms based on their distribution in the text. In a nutshell, similarities between terms can be computed on account of their co-occurrences. This is strictly connected to our supervised classifier, which is modeled in a vector space and takes into account both bag of terms and contextual windows (cf. Section 4.6.3), in a fashion similar to [1].

Matrix Factorization

Matrix factorization strategies applied to text categorization, e.g., [128], are shown to increase the performance of SVM classifiers, which we exploit:

[11] https://en.wikipedia.org/wiki/Supervised_learning

the key idea involves the construction of latent feature spaces, thus being closely related to Latent Semantic Indexing [36] techniques. While this line of work differs from ours, we believe it could be useful to optimize the features we use in the supervised classification setting.

Semantic Role Labeling

In broad terms, the Semantic Role Labeling (SRL) NLP task targets the identification of arguments attached to a given predicate in natural language utterances. From a Frame Semantics perspective, such activity translates into the assignment of FEs. This applies to efforts such as [67], and tools like MATE [13], while we perform full frame classification. On the other hand, systems like SEMAFOR [71, 32] also serve the frame disambiguation part, uniformly to our method. Hence, SEMAFOR could be regarded as a baseline system. Nonetheless, it was not possible to actually perform a comparative evaluation of our use case in Italian, since the parser exclusively supports the English language.

All the work mentioned above (and SRL in general) builds upon preceding layers of NLP machinery, i.e., POS-tagging and syntactic parsing: the importance of the latter is especially stressed in [107], thus being in strong contrast to our approach, where we propose a full bypass of the expensive syntactic step.

2.5 Taxonomy Learning

Taxonomy learning is the process of automatically inducing a hierarchy of concepts from unstructured or semi-structured data. The long thread of research focusing on taxonomy learning from digital documents dates back to the 1970s [17]. It is out of scope for this thesis to present an exhaustive literature review of such an extensive field of study. Instead,

we concentrate on Wikipedia-related work. Ponzetto and Strube [102, 103] have pioneered the stream of the Wikipedia category system taxonomization efforts, providing a method for the extraction of a class hierarchy out of the category graph. While they integrate rule-based and lexico-syntactic-based approaches to infer intra-categories is-a relations, they do not distinguish between actual instances and classes.

2.5.1 Wikipedia-powered Knowledge Bases

Large-scale knowledge bases are experiencing a steadily growing commitment of both research and industry communities. A plethora of resources have been released in recent years. Table 2.1 reports an alphabetically ordered summary of the most influential examples, which all attempt to extract structured data from Wikipedia, although with different aims.

Table 2.1: Overview of Wikipedia-powered knowledge bases (*C*ategories, *P*ages, *M*ultilingual, 3^{rd}party data). ♢ indicates a caveat

Resource	C	P	M	3
BabelNet [87]	✔	✔	✔	✔
DBpedia [73, 12]	✔	✔	✔	✔
Freebase [14]	✘	✔	✔♢	✔
MENTA [34]	✔	✔	✔	✘
WiBi [49]	✔	✔	✘	✘
Wikidata [125]	✔	✔	✔	✔
WikiNet [85, 86]	✔	✔	✔	✘
WikiTaxonomy [102, 103]	✔	✘	✘	✘
YAGO [118, 63]	✔	✔	✘	✔

BabelNet [87] is a multilingual lexico-semantic network, which recently moved towards a Linked Data compliant representation [41]. It provides wide-coverage lexicographic knowledge in 50 languages, where common concepts and real-world entities are linked together via semantic relations.

Under this perspective, BabelNet emanates from the lexical databases community, with WordNet [45] being the most mature approach. In contrast to our work, priority is given to fine-grained conceptual completeness, rather than cognitively intuitive knowledge representation. DBpedia [73, 12] leads current approaches based on the automatic extraction of unstructured and semi-structured content from all the Wikipedia language chapters. It serves as the kernel of the Linked Data cloud, gathering a huge amount of research efforts in the Web of Data and Natural Language Processing. The underlying framework is strengthened by a vibrant open source community of users and developers. However, the current paradigm employed for the ontology weakens the data consumption capabilities. Freebase [14] is the result of a crowdsourced effort, bearing a fine-grained schema thanks to its contributors. Nevertheless, no type hierarchy exists: the collaborative paradigm has actually been privileged to logical consistency. Furthermore, multilingualism is biased towards English (cf. the $\diamondsuit$ symbol in Table 2.1), since information in other languages only appears when a Wikipedia page has an English counterpart. MENTA [34] is a massive lexical knowledge base, with data coming from 271 languages. The taxonomy extraction is carried out via supervised techniques, based on a manually annotated training phase, which diminishes the replicability potential, as opposed to our fully unsupervised method. Wikidata [125] stems from the Wikimedia Foundation and is the official Wikipedia sister project. Its data model differs from all the reviewed resources, since it favors plurality over authority, in a completely collaborative fashion. It builds upon claims instead of assertions, encapsulating both temporal and provenance aspects of a given fact. The schema is crowdsourced as in Freebase. WiBi [49] attempts to produce a double taxonomy by taking into account Wikipedia knowledge encoded both at the category and at the page layers. This is in clear contrast with our work, which concentrates on the category layer to construct a

classification backbone for the page layer. Similarly to us, it does not leverage third party resources and is implemented under an unsupervised paradigm. WikiNet [85, 86] is built on top of heuristics formulated upon the analysis of Wikipedia content to deliver a multilingual semantic network. Besides is-a relations, like we do, it also learns other kinds of relations. While it seems to attain wide coverage, a comparative evaluation performed in [49] highlights very low precision.

The approach that most influenced our work is YAGO [63, 118]. Its main purpose is to provide a linkage facility between categories and WordNet terms. Conceptual categories (e.g. PERSONAL WEAPONS) serve as class candidates and are separated from administrative (e.g. CATEGORIES REQUIRING DIFFUSION), relational (e.g. 1944 DEATHS) and topical (e.g. MEDICINE) ones. Similarly to us, linguistic-based processing is applied to isolate conceptual categories.

2.5.2 Type Inference

On the other hand, the recently proposed automatic methods for type inference [94, 5, 99, 51] have yielded resources that may enrich, cleanse or be aligned to DBPO's class hierarchy. Moreover, they can serve as an assisting tool to prevent redundancy, namely to alert a human contributor when he or she is trying to add some new class that already exists or has a similar name. Hence, these efforts represent alternative solutions compared to our work, with Tìpalo [51] being the most related one.

2.6 Recommender Systems

Given a set of input items, a recommender system is a tool that suggests additional relevant ones to an end user. Current approaches merge different algorithms: the mostly exploited ones are collaborative filtering (CF),

and content-based recommendation (CB). The former typically computes suggestions based on user profiles mining, while the latter leverages bag-of-words content analysis.

Most of the work in the recommender system research community operates in a space where both item and user profiles are taken into account. We are aware that our approach must implement user profiling algorithms in order to be compared to state-of-the-art systems. So far, we have derived the following assumptions from empirical observations on our use case. (a) Post-click news recommendation generally relies on scarce user data, namely an implicit single click which can be difficult to interpret as a preference. (b) News content experiences a regular update flow, where items are not likely to be already judged by users. In such a scenario, it is known that collaborative filtering (CF) algorithms are not suitable [65, 114, 75, 21]. Instead, content-based ones (CB) apply to unstructured text, thus fitting to news articles. Document representation with bag-of-words vector space models and the cosine similarity function still represent a valid starting point to suggest topic-related documents [95]. Nevertheless, CB is concerned by the overspecialization problem, which may frustrate users [80, 76] because of recommendation sets with too similar items. Moreover, both CF and CB strategies are affected by cold-start [65, 114, 75, 21, 74], namely when new users with no profile data are recommended new items. Hence, we currently concentrate our research on investigating the role of large-scale structured knowledge bases in the CB recommendation process.

2.6.1 CF and CB systems

Although CF performs effectively when enough user data is available [18], it is affected by known limitations [65, 75] including (a) data sparsity, (b) the new item and (c) the new user problems, and (d) the lack of recommendation explanation. Content analysis techniques allow CB to tackle

typical CF problems. The active user profile is sufficient to compute recommendations and does not require neighbors, namely users with similar interests who have provided rating data (a). New items that are not rated yet can be recommended (b). Features extracted from item descriptions enable the construction of explanation systems (d). However, the overspecialization [75, 18] and portfolio [18] effects are key issues for CB, as they lead to recommendations that are too similar to a user's long term preferences (history) or to one another, thus creating a "more of the same" problem. In addition, user preferences analysis is still required, therefore (c) is not resolved. Ultimately, content analysis is inherently limited by the amount of information included in each item description. Keyword-driven algorithms usually do not consider the semantics hidden in natural language discourse. Consequently, external knowledge is often needed to improve both user tastes interpretation and items representation. Semantic-boosted approaches linking raw text documents to ontologies such as WordNet[12] or large-scale knowledge repositories such as Wikipedia have recently emerged. A literature review in this area is out of the scope of this thesis.

2.6.2 Similarity, diversity, coherence

The insertion of diverse recommendations may overcome the overspecialization problem, thus improving the quality of the system. Diversity includes novelty, namely an unknown item that a user might discover by him or herself, and serendipity, namely a completely unexpected but interesting item. While generic diversity can be assessed in terms of dissimilar items within the recommendation set via standard experimental measures, serendipity evaluation requires real user feedback, due to its subjective nature [80]. The experiments described in [122] show that serendipitous information filtering, namely the dynamic generation of suggestion lists, enhances the

[12] http://wordnet.princeton.edu

attractive power of an information retrieval system. Nevertheless, rendering such intuition into a concrete implementation remains an open problem. Randomness, user profiling, unrelatedness and reasoning by analogy are proposed starting points. Coherence in a chain of documents is another factor contributing to the quality of recommendations [76, 113].

2.6.3 Linked Open Data for recommendation

Knowledge extraction from structured data for recommendation enhancement is an attested strategy. LOD datasets, e.g., DBpedia and Freebase, are queried to enrich with properties the entities extracted from news articles [72], to collect movie information for movies recommendations [37, 121], or to suggest music for photo albums [26]. Structured data may be also mined in order to compute similarities between items, then between user and items [65].

2.6.4 Use of semantic networks for news recommendation

Formal conceptual models (ontologies) are known to improve user and item profiling, as they alleviate keyword-based approaches problems by injecting semantics [75]. [21, 19, 20] leverage semantic relations within the user per item space for a news recommender system. Annotations extracted from news articles (semantic context) enrich a pre-existing ontology to achieve more complex and disambiguated item/user representations. However, such annotations only originate from news titles and summaries, not from the whole textual content. Moreover, natural language processing techniques used for text annotation do not take into account state-of-the-art entity linking tools [82], based on machine learning and word sense disambiguation algorithms, e.g., [**?**]. Recommendations are finally ranked via a cosine similarity score between user preferences and item annotations vectors.

[72] proposes a hybrid news articles recommendation system, which merges content processing techniques and data enrichment via LOD. This approach is similar to ours with respect to the article processing: offline corpus gathering, named entity extraction and LOD exploitation. A document is modeled with traditional information retrieval measures such as TF-IDF weights for terms and an adaptation of the formula for named entities, which basically substitutes the term frequency with a normalized entity frequency. Natural language subject-verb-object sentences e.g. `Microsoft recommends reinstalling Windows` are also taken into account.

[114] exploit an ontology classifying both user and item profiles for a personalized newspaper. A common conceptual representation improves the computation of relevant items to a given user. Similarly to our entity linking and schema inspection steps[13], an item profile is described by a set of representative ontology terms. A user profile is initially constructed via explicitly selected interests from the ontology terms and is maintained by implicit feedback. When a user has clicked on a new item, the associated terms are updated to his or her profile as new interests. The authors assume here that a click on a news item corresponds to a positive preference, which may bias the profile. The similarity between an item and a user is based on the weighted number of perfect or partial matches between the terms describing that item and the terms describing that user, and yields a ranking score for the final recommendations.

2.6.5 Other approaches for news recommendation

[113] describe a method for producing a coherent path (story) between two news articles. The authors list the drawbacks of keyword-driven approaches, namely the creation of weak links based on word overlap, the loss of potentially significant features due to the absence of certain

[13] Detailed in section 6.3.

words in a given article, and the non-consideration of word importance. The proposed solution incorporates the influence score of a document on another through an activated concept, namely word patterns that are shared among documents. While this is comparable to our relation discovery between entities of a document, the exploitation of external knowledge to establish the connections is not taken into account, since the triggering patterns remain in the document space. The presented algorithm suffers from scaling issues. A possible solution could be the pre-selection of both document and concept subsets. Finally, the tradeoff between relevance and redundancy is pointed out. Overall relevance can be improved by injecting more similar documents in the chain, although increasing redundancy.

[74] represent the problem of news recommendation as a contextual multi-armed bandit problem. As mentioned above, a news recommender system must cope with constant data updates and a cold-start scenario. Hence, it should be able to rapidly select interesting articles for upcoming users. Different multi-armed bandit techniques attempt to handle cold-start. (a) Context-free algorithms disregard both user and item features. (b) Warm start algorithms infer personalization offline from overall click-through rates (CTR). (c) Contextual algorithms dynamically learn from user-centric CTRs. Given a set of arms, namely the candidate articles, the proposed bandit algorithm tries to guess the best arm based on previously gathered payoffs, namely the users' CTR on that article. In each trial, contextual information is represented as a vector of features containing both the current user and the arm profiles. When an arm is selected (i.e., an article is shown), payoffs of 1 or 0 are collected (i.e., if the article is clicked or not). Ultimately, the algorithm refines its arm choice thanks to the acquired payoffs. Since the maximum payoff of an arm corresponds to the maximum CTR of an article, the strategy is able to promptly recognize potentially attractive articles for an unseen user. However, the authors

narrow their focus on the algorithm computational efficiency. Moreover, user/item feature vectors are constructed upon explicit user profiles, reading histories and manually annotated article categories. They claim that user information is commonly available in web services and can be consumed to build user feature vectors. Besides such claim does not necessarily apply to all news portals, the approach still depends on explicit user data in order to generalize CTR and compute recommendations. Finally, the correct interpretation of implicit click feedbacks remains an open problem.

2.6.6 Evaluation guidelines

Recommender systems evaluation frameworks boil down to two main approaches [65], namely (a) offline and (b) online. (a) leverages gold-standard datasets and aims at estimating the performance of a recommendation algorithm via statistical measures. (b) relies on real user studies. [129] adopt both approaches. [74] demonstrate how to evaluate all bandit algorithms offline with web logs. While [18] used ad-hoc created datasets, in [21] the authors claim that their algorithms need to be shaped on such data, thus restraining the evaluation capabilities. Therefore, they performed an online evaluation. The difficulty to provide explicit ratings for some items and the permanence of the overspecialization issue only emerged thanks to a set of evaluators' comments. [80] highlight that the priority accorded to offline accuracy measures has negatively biased system evaluations with respect to end users' perspective. [58] argue that user satisfaction corresponds to the actual use of a system and can be effectively measured only via online evaluation. The interest in exploiting crowdsourcing services for dataset building and online evaluation has recently grown, especially with respect to natural language processing tasks [88] and behavioral research [78].

Chapter 3

Crowdsourcing Frame Annotation

3.1 Introduction

Annotating Frame Semantics[1] information is a complex task, usually modeled in two steps: first annotators are asked to choose the situation (or *frame*) evoked by a given predicate (the *lexical unit, LU*) in a sentence, and then they assign the semantic roles (or *frame elements, FEs*) that describe the participants typically involved in the chosen frame. For instance, the sentence `Karen threw her arms round my neck, spilling champagne everywhere` contains the LU *throw*.v evoking the frame BODY_MOVEMENT. However, *throw*.v is ambiguous and may also evoke CAUSE_MOTION. Existing frame annotation tools, such as SALTO [16] and the BERKELEY system [48] foresee this two-step approach, in which annotators first select a frame from a large repository of possible frames (1,162 frames are currently listed in the online version of the resource), and then assign the FE labels constrained by the chosen frame to LU dependents.

In this chapter, we argue that such workflow shows some redundancy which can be addressed by radically changing the annotation methodology and performing it in one single step. Our novel annotation approach is also more compliant with the definition of frames proposed in [47]: in his seminal

[1]The reader may refer to Section 2.2 for a detailed description of the theory.

work, Fillmore postulated that the meanings of words can be understood on the basis of a semantic frame, i.e., a description of a type of event or entity and the participants in it. This implies that frames can be distinguished one from another on the basis of the participants involved, thus it seems more cognitively plausible to start from the FE annotation to identify the frame expressed in a sentence, and not the contrary.

The goal of our methodology is to provide full frame annotation in a single step and in a bottom-up fashion. Instead of choosing the frame first, we focus on FEs and let the frame emerge based on the chosen FEs. We believe this approach complies better with the cognitive activity performed by annotators, while the 2-step methodology is more artificial and introduces some redundancy because part of the annotators' choices are replicated in the two steps (i.e. in order to assign a frame, annotators implicitly identify the participants also in the first step, even if they are annotated later).

Another issue we investigate in this work is how semantic roles should be annotated in a crowdsourcing framework. This task is particularly complex, therefore it is usually performed by expert annotators under the supervision of linguistic experts and lexicographers, as in the case of FrameNet. In NLP, different annotation efforts for encoding semantic roles have been carried out, each applying its own methodology and annotation guidelines (see for instance [111] for FrameNet and [93] for PropBank). In this work, we present a pilot study in which we assess to what extent role descriptions meant for 'linguistics experts' are also suitable for annotators from the crowd. Moreover, we show how a simplified version of these descriptions, less bounded to a specific linguistic theory, improve the annotation quality.

3.2 Experiments

In this section, we describe the anatomy and discuss the results of the tasks we outsourced to the crowd via CrowdFlower. Before diving into them, we report a set of critical aspects underpinning the platform.

Golden Data. Quality control of the collected judgements is a key factor for the success of the experiments. The essential drawback of crowdsourcing services relies on the cheating risk. Workers are generally paid a few cents for tasks which may only need a single click to be completed. Hence, it is highly probable to collect data coming from random choices that can heavily pollute the results. The issue is resolved by adding *gold* units, namely data for which the requester already knows the answer. If a worker misses too many gold answers within a given threshold, he or she will be flagged as untrusted and his or her judgments will be automatically discarded.

Worker Switching Effect. Depending on their accuracy in providing answers to gold units, workers may switch from a trusted to an untrusted status and vice versa. In practice, a worker submits his or her responses via a web page. Each page contains one gold unit and a variable number of regular units that can be set by the requester during the calibration phase. If a worker becomes untrusted, the platform collects another judgment to fill the gap. If a worker moves back to the trusted status, his or her previous contribution is added to the results as free extra judgments. Such phenomenon typically occurs when the complexity of gold units is high enough to induce low agreement in workers' answers. Thus, the requester is constrained to review gold units and to eventually forgive workers who missed them.

Cost Calibration. The total cost of a crowdsourced task is naturally bound to a data unit. This represents an issue in our experiments, as the number of questions per unit (i.e. a sentence) varies according to the number of frames and FEs evoked by the LU contained in a sentence. Therefore, we need to use the average number of questions per sentence as a multiplier to a constant cost per sentence. We set the payment per working page to 5 $ cents and the number of sentences per page to 3, resulting in 1.83 $ cent per sentence.

3.2.1 Assessing Task Reproducibility and Worker Behavior Change

Since our overall goal is to compare the performance of FrameNet annotation using our novel workflow to the performance of the standard, 2-step approach, we first take into account past related works and try to reproduce them. To our knowledge, the only attempt to annotate frame information through crowdsourcing is the one presented in [64], which however did not include FE annotation.

Modeling. The task is designed as follows. (a) Workers are invited to read a sentence where a LU is bolded; (b) the question *"Which is the correct sense?"* is combined with the set of frames evoked by the given LU, as well as the `None` choice; finally, (c) workers must select the correct frame. A set of example sentences corresponding to each possible frame is provided in the instructions to facilitate workers. For instance, the sentence *"Leonardo Di Caprio **won** the Oscar in 2016"* is displayed with the set of frames WIN_PRIZE, FINISH_COMPETITION, GETTING, FINISH_GAME triggered by the LU *win*.v, together with `None`. The worker should pick WIN_PRIZE.

As a preliminary study, we wanted to assess to what extent the proposed task could be reproduced and if workers reacted in a comparable way over time. [64] did not publish the input datasets, thus we ignore which sentences

LU	2013		2011
	Sentences (Gold)	Accuracy	Accuracy
high.a	68 (9)	91.8	92
history.n	72 (9)	84.6	86
range.n	65 (8)	95	93
rip.v	88 (12)	81.9	92
thirst.n	29 (4)	90.4	95
top.a	36 (5)	98.7	96

Table 3.1: Comparison of the reproduced frame discrimination task as per [64]

were used. Besides, the authors computed accuracy values directly from the results upon a majority vote ground truth. Therefore, we decided to consider the same LUs used in Hong and Baker's experiments, i.e., *high*.a, *history*.n, *range*.n, *rip*.v, *thirst*.n and *top*.a, but we leveraged the complete sets of FrameNet 1.5 expert-annotated sentences as gold-standard data for immediate accuracy computation.

Discussion. Table 3.1 displays the results we achieved, jointly with the experiments by [64]. For the latter, we only show accuracy values, as the number of sentences was set to a constant value of 18, 2 of which were gold. If we assume that the crowd-based ground truth in 2011 experiments is approximately equivalent to the expert one, workers seem to have reacted in a similar manner compared to Hong and Baker's values, except for *rip*.v.

3.2.2 General Task Setting

We randomly chose the following LUs among the set of all verbal LUs in FrameNet evoking 2 frames each: *disappear.v* [Ceasing_to_be, Departing], *guide.v* [Cotheme, Influence_of_event_on_cognizer], *heap.v* [Filling, Placing], *throw.v* [Body_movement, Cause_motion]. We

considered verbal LUs as they usually have more overt arguments in a sentence, so that we were sure to provide workers with enough candidate FEs to annotate. Linguistic tasks in crowdsourcing frameworks are usually decomposed to make them accessible to the crowd. Hence, we set the polysemy of LUs to 2 to ensure that all experiments are executed using the smallest-scale subtask. More frames can then be handled by just replicating the experiments.

3.2.3 2-step Approach

After observing that we were able to achieve similar results on the frame discrimination task as in previous work, we focused on the comparison between the 2-step and the 1-step frame annotation approaches.

We first set up experiments that emulate the former approach both in frame discrimination and FEs annotation. This will serve as the baseline against our methodology. Given the pipeline nature of the approach, errors in the frame discrimination step will affect FE recognition, thus impacting on the final accuracy. The magnitude of such effect strictly depends on the number of FEs associated with the wrongly detected frame.

Frame Discrimination. Frame discrimination is the first phase of the 2-step annotation procedure. Hence, we need to leverage its output as the input for the next step.

Modeling The task is modeled as per Section 3.2.1.

Discussion Table 6.4.3 gives an insight into the results, which confirm the overall good accuracy as per the experiments discussed in Section 3.2.1.

Frame Elements Recognition. We consider all sentences annotated in the previous subtask with the frame assigned by the workers, even if it is not correct.

Modeling. The task is presented as follows. (a) Workers are invited to read a sentence where a LU is bolded and the frame that was identified in the first step is provided as a title. (b) A list of FE definitions is then shown together with the FEs text chunks. Finally, (c) workers must match each definition with the proper FE.

Approach	2-STEP		1-STEP
Task	FD	FER	
Accuracy	.900	.687	**.792**
Answers	100	160	416
Trusted	100	100	84
Untrusted	21	36	217
Time (h)	102	69	**130**
Cost/question ($ cents)	1.83	2.74	8.41

Table 3.2: Overview of the experimental results. FD stands for Frame Discrimination, FER for FEs Recognition

Simplification. Since FEs annotation is a very challenging task, and FE definitions are usually meant for experts in linguistics, we experimented with three different types of FE definitions: the original ones from FrameNet, a manually simplified version, and an automatically simplified one, using the tool by [59]. The latter simplifies complex sentences at the syntactic level and generates a question for each of the extracted clauses. As an example, we report below three versions obtained for the *Agent* definition in the DAMAGING frame:

Original: The conscious entity, generally a person, that performs the intentional action that results in the damage to the Patient.

Manually simplified: This element describes the person that performs the intentional action resulting in the damage to another person or object.

Automatic system: What that performs the intentional action that results in the damage to the Patient?

Simplification was performed by a linguistic expert, and followed a set of straightforward guidelines, which can be summarized as follows:

- When the semantic type associated with the FE is a common concept (e.g. `Location`), replace the FE name with the semantic type.
- Make syntactically complex definitions as simple as possible.
- Avoid variability in FE definitions, try to make them homogeneous (e.g. they should all start with "This element describes..." or similar).
- Replace technical concepts such as `Artifact` or `Sentient` with common words such as `Object` and `Person` respectively.

Although these changes (especially the last item) may make FE definitions less precise from a lexicographic point of view (for instance, sentient entities are not necessarily persons), annotation became more intuitive and had a positive impact on the overall quality.

After few pilot annotations with the three types of FE definitions, we noticed that the simplified one achieved a better accuracy and a lower number of untrusted annotators compared to the others. Therefore, we use the simplified definitions in both the 2-step and the 1-step approach (Section 3.2.4).

Discussion. Table 6.3 provides an overview of the results we gathered. The total number of answers differs from the total number of trusted judgments,

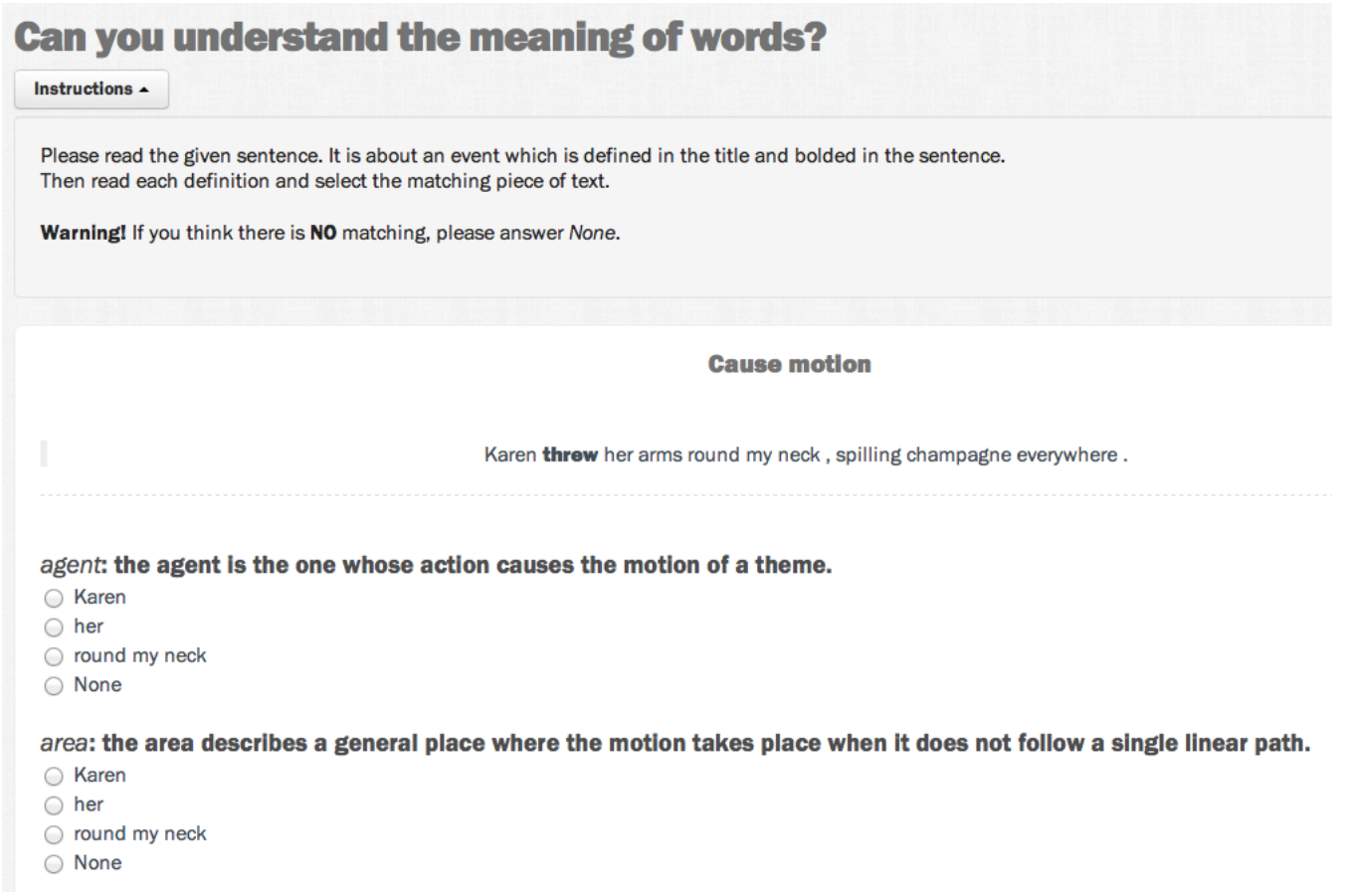

Figure 3.1: 1-step approach worker interface

since the average value of questions per sentence amounts to 1.5.[2] First of all, we notice an increase in the number of untrusted judgments. This is caused by a generally low inter-worker agreement on gold sentences due to FE definitions, which still present a certain degree of complexity, even after simplification. We inspected the full reports sentence by sentence and observed a propagation of incorrect judgments when a sentence involves an unclear FE definition. As FE definitions may mutually include mentions of other FEs from the same frame, we believe this circularity generated confusion.

[2]Cf. Section 3.2 for more details

3.2.4 1-step Approach

Having set the LU polysemy to 2, in our case a sentence S always contains a LU with 2 possible frames (f_1, f_2), but only conveys one, e.g., f_1. We formulate the approach as follows. S is replicated in 2 data units (S_a, S_b). Then, S_a is associated to the set E_1 of f_1 FE definitions, namely the correct ones for that sentence. Instead, S_b is associated to the set E_2 of f_2 FE definitions. We call S_b a *cross-frame* unit. Furthermore, we allow workers to select the `None` answer. In practice, we ask a total amount of $|E_1 \cup E_2| + 2$ questions per sentence S. In this way, we let the frame directly emerge from the FEs. If workers correctly answer `None` to a FE definition $d \in E_2$, the probability that S evokes f_1 increases.

Modeling. Figure 3.4 displays a screenshot of the worker interface. The task is designed as per Section 3.2.3, but with major differences with respect to its content. For instance, given the running example introduced in Section 3.1, we ask to annotate both the BODY_MOVEMENT and the CAUSE_MOTION core FEs, respectively as regular and cross-frame units.

Discussion. We do not interpret the `None` choice as an abstention from judgment, since it is a correct answer for cross-frame units. Instead of precision and recall, we are thus able to directly compute workers' accuracy upon a majority vote. We envision an improvement with respect to the 2-step methodology, as we avoid the proven risk of error propagation originating from wrongly annotated frames in the first step. Table 6.3 illustrates the results we collected. As expected, accuracy reached a consistent enhancement. This demonstrates the hypothesis we stated in Section 3.1 on the cognitive plausibility of a bottom-up approach for frame annotation. Furthermore, the execution time decreases compared to the sum of the 2 steps, namely 130 hours against 171. Nevertheless, the cost is sensibly higher due to the

higher number of questions that need to be addressed, in average 4.6 against 1.5. Untrusted judgments seriously grow, mainly because of the cross-frame gold complexity. Workers seem puzzled by the presence of `None`, which is a required answer for such units. If we consider the English FrameNet annotation agreement values between experts reported by [92] as the upper bound (i.e., .897 for frame discrimination and .949 for FEs recognition), we believe our experimental setting can be reused as a valid alternative.

3.3 Improving FEs Annotation with DBpedia

Since we aim at investigating whether such activity can be cast to a crowd of non-expert contributors, we need to reduce its complexity by intervening on the FE descriptions. In particular, we want to assess to what extent more information on the role semantics coming from external knowledge sources such as DBpedia can improve non-expert annotators' performance. We claim that providing annotators with information on the semantic types typically associated with FEs will enable faster and cheaper annotations, while maintaining an equivalent accuracy. The additional information is extracted in a completely automatic way, and the workflow we present can be potentially applied to any crowdsourced annotation task in which semantic typing is relevant.

3.3.1 Annotation Workflow

Our goal is to determine if crowdsourced annotation of semantic roles can be improved by providing non-expert annotators with information from DBpedia on the roles they are supposed to label. Specifically, instead of displaying the lexicographic definition for each possible role to be labeled, annotators are shown a set of semantic types associated with each role coming from FrameNet. Based on this, annotators should better recognize

such roles in an unseen sentence. Evaluation is performed by comparing this annotation framework with a baseline, where standard FE definitions substitute DBpedia information.

Before performing the annotation task, we need to leverage the list of semantic types that best characterizes each FE in a frame. We extract these statistics by connecting the FrameNet database 1.5 [111] to DBpedia, after isolating a set of sentences to be used as test data (cf. Section 3.4). The workflow to prepare the input for the crowdsourced task is based on the following steps.

Linking to Wikipedia

For each annotated sentence in the FrameNet database, we first link each textual span labeled as FE to a Wikipedia page W. We employ The Wiki Machine, a kernel-based linking system (details on the implementation are reported in [123, 54]), which was trained on the Wikipedia dump of March 2010.[3] Since FEs can be expressed by both common nouns and real-world entities, we needed a linking system that satisfactorily processes both nominal types. A comparison with the state-of-the-art system Wikipedia Miner [83] on the ACE05-WIKI dataset [10] showed that The Wiki Machine achieved a suitable performance on both types (.76 F1 on real-world entities and .63 on common nouns), while Wikipedia Miner had a poorer performance on the second noun type (respectively .76 and .40 F1). These results were also confirmed in a more recent evaluation [82], in which The Wiki Machine achieved the highest F1 compared with an ensemble of academic and commercial systems, such as *DBpedia Spotlight*, *Zemanta*, *Open Calais*, *Alchemy API*, and *Ontos*.

The system applies an *all-word* linking strategy, in that it tries to connect each word (or multiword) in a given sentence to a Wikipedia page. In case

[3] http://download.wikimedia.org/enwiki/20100312

a linked textual span (partially) matches a string corresponding to a FE, we assume that one possible sense of FE is represented in Wikipedia through W. The Wiki Machine also assigns a confidence score to each linked term. This confidence is higher in case the words occurring in the same context of the linked term show high similarity, because the system considers that the linking is likely to be more accurate.

We illustrate in Figure 3.2 the Wikipedia pages (and confidence score) that the Wiki Machine system associates with the sentence `Sardar Patel was assisting Gandhiji in the Salt Satyagraha with great wisdom`, an example sentence for the ASSISTANCE frame originally annotated with four FEs, namely Helper, Benefited_party, Goal and Manner. Since Wikipedia is a repository of concepts, which are usually expressed by nouns, we are able to link only nominal fillers.

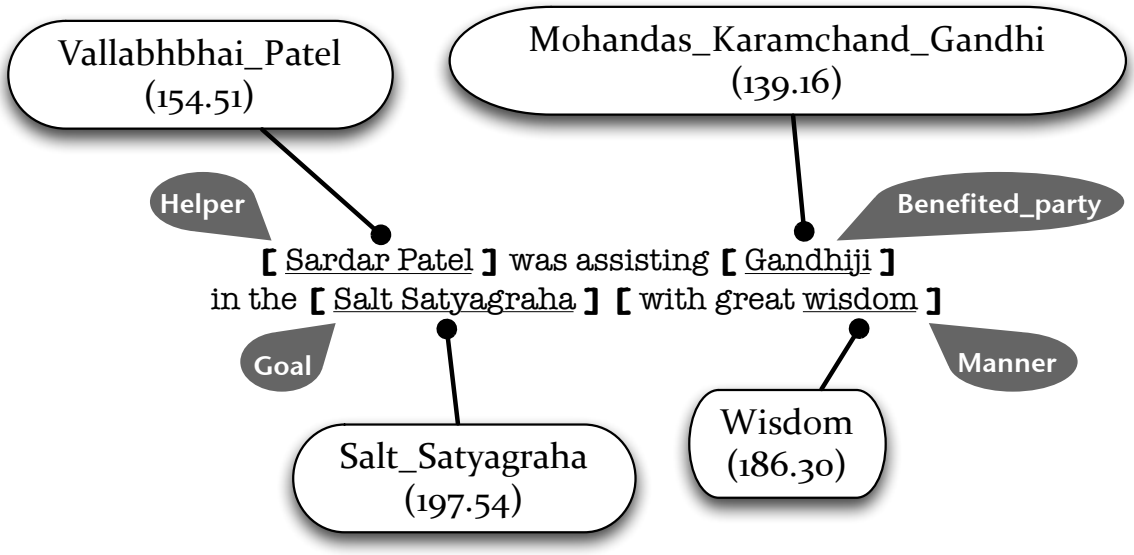

Figure 3.2: Linking example with confidence score

Linking to DBpedia

In order to obtain the semantic types that are typical for each FE, linking to Wikipedia is not enough. In fact, too many different pages would be connected to a FE, making it difficult to generalize over the Wikipedia pages (i.e. concepts). This emerges also from the example above, where the pages linked to `Sardar Patel`, `Gandhjii` and `Salt Satyagraha` do not provide information on the typical fillers of Helper, Benefited_party and Goal respectively. One possible option could be to resort to Wikipedia categories, which however are not homogenous enough to allow for a consistent extraction of FE semantic types.

We tackle this problem by using Wikipedia pages as a bridge to DBpedia. In fact, Wikipedia page URLs directly map to DBpedia resource URIs. Hence, for each linked FE, we query DBpedia for `rdf:type` objects. In this way, we are able to rank the most frequent semantic types associated with a given FE from a given frame. For instance, the FE Victim from the frame KILLING would link to the DBpedia type *Animal*, which ranks first in our input data, with 38 occurrences (cf. Section 3.4). We aim at investigating whether such top-occurring types represent both valid generalizations and simplifications of a standard FE definition, and may thus substitute it. At the end of this pre-processing step, we create a repository where, for each FE, a set of DBpedia types is listed and ranked by frequency.

Posting the Annotation Task on CrowdFlower

We finally set up a crowdsourced experiment where, in each test sentence, annotators have to choose the most appropriate FE given the most frequent DBpedia types (proper task) or the standard FE definition (baseline). Details are reported in the following section.

Table 3.3: FrameNet data processing details

Workflow step	FE instances
Raw FrameNet	148,440
Linking to Wikipedia	114,242
DBpedia types extraction	47,732

3.4 Experiments

We first provide an overview of critical aspects underpinning a generic crowdsourced experiment. Subsequently, we describe the anatomy and the modeling of the tasks we outsourced to the CrowdFlower platform. The worker switching effect has not been a blocking issue in our experiments, since we assessed a relatively low average percentage of missed judgments for gold units, namely 28%. We set the payment per working page to 3 $ cents and the number of sentences per page to 3.

Pre-processing of FrameNet Data for DBpedia Types Extraction

Table 3.3 provides some statistics of the processed FrameNet data that were leveraged to extract DBpedia types (cf. Section 3.3.1). More specifically:

1. From the FrameNet 1.5 database, the Wiki Machine managed to link 77% of the total number of FE instances. Hence, unlinked data is skipped for the next step.

2. DBpedia provided type information for 42% of the total number of linked FE instances. Types occurring once are ignored, as they reflect the content of a single sentence and are likely to convey misleading suggestions. The too generic `owl#Thing` type is filtered as well.

Table 3.4: Experimental settings

Sentences	43
Gold	6
Frames	24
Lexical Units	41
Average FEs per sentence	3.07
Average cost per FE ($ cents)	.325
Average DBpedia types per FE	4.66
Workers nationality	United States

Test Data Preparation

Before linking the FrameNet database to DBpedia, we isolate a subset to be used as test data. From 500 randomly chosen sentences, we select those in which the number of FEs per frame is between 3 and 4.

This small dataset serves as input for our experiments. Table 3.4 details the final settings. We hand-pick six sentences and for each of them we mark one question as gold for quality check. Almost all sentences contain three FEs with few exceptions (cf. the average value in Table 3.4). We extract the five most frequent DBpedia types from the statistics and assign them to the corresponding FEs in our input. Since not all FEs have exactly five associated types (cf. the average value in Table 3.4), we provide workers with variable suggestion sets. Finally, we ensure all workers are native English speakers.

Modeling

Data units are delivered to workers via a web interface. Our task is illustrated in Figure 3.3 and is presented as follows:

(a) Workers are invited to read a sentence and to focus on the bolded word appearing as a title above the sentence (e.g. `taste` in the screenshot).

(b) A question concerning each FE is then shown together with a set of answers corresponding to the sentence chunks that may express the given FE. For instance, in Figure 3.3, the question `Which is the Perceiver Passive?` is coupled with multiple choices taken from the given sentence.

(c) For each question, a suggestion box displays the top types retrieved from DBpedia and connected to the given FE (cf. Section 3.3.1 for details). This should help annotators in choosing the text chunk that better fits the given FE.

(d) Finally, workers match each question with the proper text chunk.

On the other hand, the baseline differs from our strategy in that (i) it does not display the suggestion box and (ii) questions are replaced with the FE definition extracted from FrameNet. For instance, in Figure 3.3, the question about the Perceiver Passive would be replaced with `This FE is the being who has a perceptual experience, not necessarily on purpose`. The baseline is more compliant with the standard approach adopted to annotate FEs in the FrameNet project.

Figure 3.3: Worker interface unit screenshot

3.5 Results

Our main purpose is to evaluate the validity of the proposed approach against the conventional FrameNet annotation procedure. We leverage expert-annotated sentences and are thus able to directly measure workers' accuracy. Specifically, we compute 2 values:

- *Majority vote.* An answer is considered correct only if the majority of judgments are correct.
- *Absolute.* The total number of correct judgments divided by the total number of collected judgments.

The results of our experiments are detailed in Table 3.5. The number of untrusted judgments may be considered as a shallow indicator of the overall task complexity. In fact, we tried to maximize objectivity and simplicity when choosing gold units. Moreover, the input dataset (and gold units as well) is identical in both experiments. Therefore, we can infer that the number of workers who missed gold is directly influenced by the question model, which is the only variable parameter. We compute the execution time as the interval between the first and the last judged unit.

Table 3.5: Overview of the experimental results

Measure	**Baseline**	**DBpedia**
Majority vote accuracy	.763	.803
Absolute accuracy	.646	.720
Untrusted judgments	90	82
Time (minutes)	160	106

Our approach outperformed the baseline both in terms of accuracy and time. While majority vote accuracy values differ slightly, absolute accuracy clearly favors our strategy. Such measure can be seen as a further indicator of the task complexity. A higher score implies a higher number of

correct judgments, which may designate a better inter-worker agreement, thus a more straightforward task. This claim is not only supported by the moderate decrease of untrusted judgments, but also by the dramatic reduction of the execution time. Consequently, the results we obtained demonstrate that entity linking techniques combined with DBpedia types simplify FEs annotation.

3.6 Conclusion

In this chapter, we first presented an approach to perform full frame annotation with crowdsourcing techniques, based on a single annotation step and on manually simplified FE definitions. Since the results of such baseline seem promising, we developed an additional method leveraging information extracted from DBpedia. The task is simplified for non-expert annotators by replacing FE definitions, usually meant for linguistic experts, with semantic types obtained from DBpedia. This is accomplished without manual simplification, in a completely automatic fashion. Results prove that such method improves on the previous annotation workflow, both in terms of accuracy and of time consumption. Although the interconnection between FEs and DBpedia is semantically not perfect, extracting frequency statistics from the whole FrameNet database and considering only the most occurring types from DBpedia make the procedure quite robust to wrong links.

Chapter 4

Properties: N-ary Relation Extraction from Free Text

4.1 Introduction

Intelligent Web-reading Agents, are Artificial Intelligence systems that can read and comprehend human language in documents across the Web. Ideally, these agents should be robust enough to interchange between heterogeneous sources with agility, while maintaining equivalent reading capabilities. More specifically, given a set of input corpora (where an item corresponds to the textual content of a Web source), they should be able to navigate from corpus to corpus and to extract comparable structured assertions out of each one. Ultimately, the collected data would feed a target Knowledge Base (KB).

In this scenario, the encyclopedia Wikipedia contains a huge amount of data, which may represent the best digital approximation of human knowledge. As an anecdotal yet remarkable proof, Google acquired Freebase, a Wikipedia-driven KB [14], in 2010,[1] embedded it in its Knowledge Graph,[2] and has lately opted to shut it down to the public.[3] Currently, it

[1] https://googleblog.blogspot.it/2010/07/deeper-understanding-with-metaweb.html
[2] https://www.google.com/intl/en_us/insidesearch/features/search/knowledge.html
[3] https://plus.google.com/109936836907132434202/posts/bu3z2wVqcQc

is foreseen [120] that Freebase data will eventually migrate to WIKIDATA[4] via the *primary sources* tool,[5] which aims at standardizing the flow for data donations.

The trustworthiness of a general-purpose KB like Wikidata is an essential requirement to ensure reliable (thus high-quality) content: as a support for their plausibility, data should be validated against third-party resources. Even though the Wikidata community strongly agrees on the concern,[6] few efforts have been approached towards this direction. The addition of references to external (i.e., non-Wikimedia), authoritative Web sources can be viewed as a form of validation. Consequently, such real-world setting further consolidates the need for an intelligent agent that harvests structured data from raw text and produces, e.g., Wikidata statements with reference URLs. Besides the prospective impact on the KB augmentation and quality, the agent would also dramatically shift the burden of manual data addition and curation, by pushing the (intended) fully human-driven flow towards an assisted paradigm, where automatic suggestions of pre-packaged statements just require to be approved or rejected. Figure 4.1 depicts the current state of the primary sources tool interface for Wikidata editors, which is in active development yet illustrates such future technological directions. Our system already takes part in the process, as it feeds the tool back-end.

On the other hand, the DBpedia EXTRACTION FRAMEWORK[7] is pretty much mature when dealing with Wikipedia semi-structured content like infoboxes, links and categories. Nevertheless, unstructured content (typically text) plays the most crucial role, due to the potential amount of extra knowledge it can deliver: to the best of our understanding, no efforts have

[4] https://www.wikidata.org/wiki/Wikidata:WikiProject_Freebase

[5] https://www.wikidata.org/wiki/Wikidata:Primary_sources_tool

[6] https://www.wikidata.org/wiki/Wikidata:Referencing_improvements_input, http://blog.wikimedia.de/2015/01/03/scaling-wikidata-success-means-making-the-pie-bigger/

[7] https://github.com/dbpedia/extraction-framework

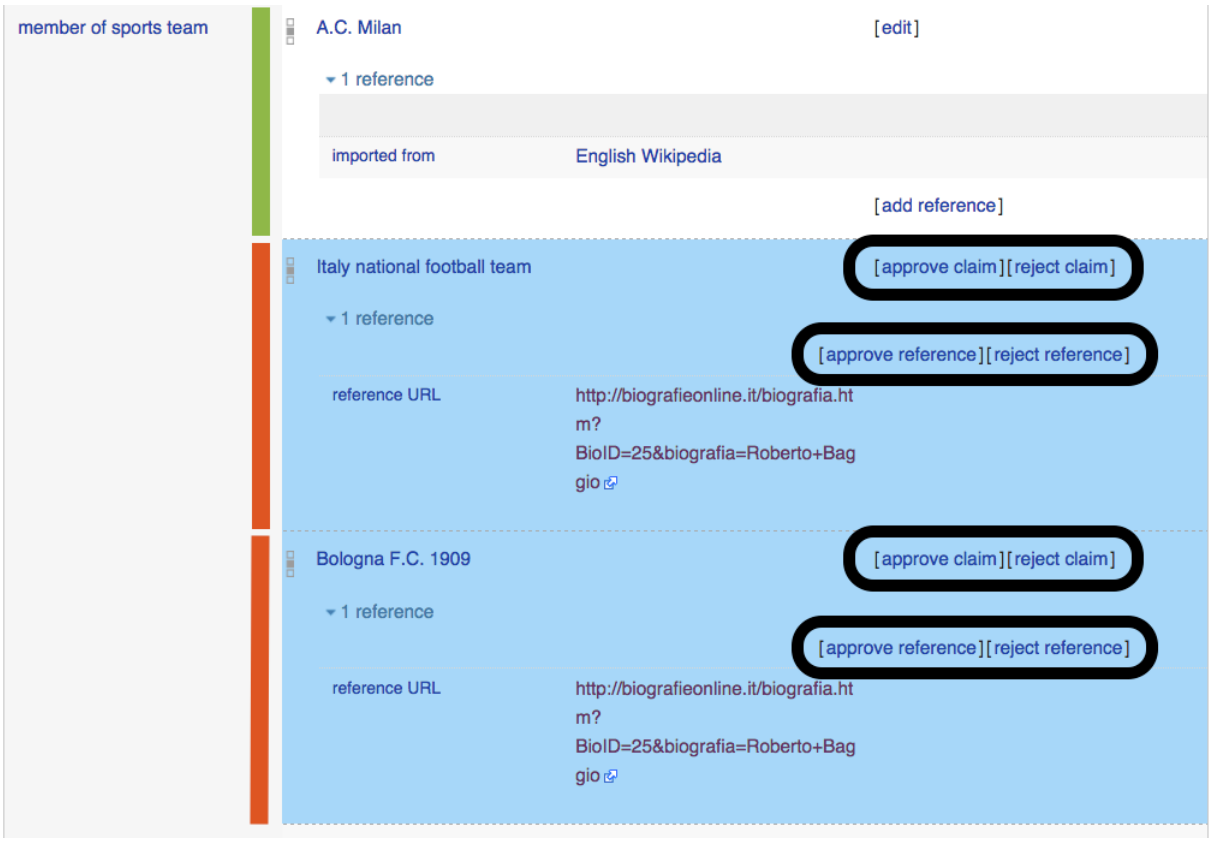

Figure 4.1: Screenshot of the Wikidata primary sources gadget activated in Roberto Baggio's page. The statement highlighted with a green vertical line already exists in the KB. Automatic suggestions are displayed with a blue background: these statements require validation and are highlighted with a red vertical line. They can be either approved or rejected by editors, via the buttons highlighted with black circles.

Table 4.1: Extraction examples on the Germany national football team article

Sentence	Extracted sta
The first manager of the Germany national team was Otto Nerz	(Germany, roster, Roster_01), (Roster
Germany has won the World Cup four times	(Germany, trophy, (Trophy_01, competition, World C
In the 70s, Germany wore Erima kits	(Germany, wearing, (Wearing_01, garment, Erima), (

been carried out to integrate an unstructured data extractor into the framework. For instance, given the Germany football team article,[8] we aim at extracting a set of meaningful facts and structure them in machine-readable statements. The sentence `In Euro 1992, Germany reached the final, but lost 0–2 to Denmark` would produce a list of *triples*, such as:

```
(Germany, defeat, Defeat_01)

(Defeat_01, winner, Denmark)

(Defeat_01, loser, Germany)

(Defeat_01, score, 0–2)

(Defeat_01, competition, Euro 1992)
```

To fulfill both Wikidata and DBpedia duties, we aim at investigating in what extent can Frame Semantics [46, 47] be leveraged to perform Information Extraction over Web documents. We foresee to exploit our novel annotation approach (cf. Chapter 3), which provides full frame annotation in a *single* step and in a bottom-up fashion (i.e., *from FEs up to frames*).

[8] http://en.wikipedia.org/wiki/Germany_national_football_team

4.1.1 Contributions

In this chapter, we focus on Wikipedia as the source corpus and on DBpedia as the target KB. We propose to apply NLP techniques to Wikipedia text in order to harvest structured facts that can be used to automatically add novel statements to DBpedia. Our Relation Extractor is set apart from related state of the art thanks to the combination of the following contributions:

1. **N-ary relation extraction**, as opposed to binary standard approaches, e.g., [44, 6, 3, 119, 43, 22], and in line with the notion of knowledge pattern [52];
2. **simultaneous T-Box and A-Box population** of the target KB, in contrast to, e.g., [40];
3. **shallow NLP machinery**, only requiring the grammatical analysis (i.e., part-of-speech tagging) layer, with no need for syntactic parsing (e.g., [79]) nor semantic role labeling (e.g., [67, 66, 71, 32, 13]);
4. **low-cost yet supervised machine learning** paradigm, via training set crowdsourcing, which ensures full supervision without the need for expert annotators.

4.1.2 Problem and Solution

The main research challenge is formulated as a KB population problem: specifically, we tackle how to automatically enrich DBpedia resources with novel statements extracted from the text of Wikipedia articles. We conceive the solution as a machine learning task implementing the Frame Semantics linguistic theory [46, 47]: we investigate how to recognize meaningful factual parts given a natural language sentence as input. We cast this as a classification activity falling into the supervised learning paradigm.

In particular, we focus on the construction of a new extractor, to be integrated into the current DBpedia infrastructure. Frame Semantics will enable the discovery of relations that hold between entities in raw text. Its implementation takes as input a collection of documents from Wikipedia (i.e., the corpus) and outputs a structured dataset composed of machine-readable statements.

The remainder of this chapter is structured as follows. We introduce a use case in Section 4.2, which will drive the implementation of our system. Its high-level architecture is then described in Section 4.3, and devises the core modules, which we detail in Section 4.4, 4.5, 4.6, 4.7, and 4.8. A baseline system is reported in Section 4.9: this enables the comparative evaluation presented in Section 8.7, among with an assessment of the T-Box and A-Box enrichment capabilities. In Section 4.11, we gather a list of research and technical considerations to pave the way for future work, before our conclusions are drawn in Section 5.7.

4.2 Use Case

Soccer is a widely attested domain in Wikipedia: according to the Italian DBpedia,[9] the Italian Wikipedia counts a total of 59,517 articles describing soccer-related entities, namely 2.63% of the whole chapter. Moreover, infoboxes on those articles are generally very rich (cf. for instance the Germany national football team article). On account of these observations, the soccer domain properly fits the main challenge of this effort. Table 4.1 displays three examples of candidate statements from the Germany national football team article text, which do not exist in the corresponding DBpedia resource. In order to facilitate the readability, the examples stem from the English chapter, but also apply to Italian.[10]

[9] As per the 2015 release, based on the Wikipedia dumps from January 2015.

[10] https://it.wikipedia.org/wiki/Nazionale_di_calcio_della_Germania

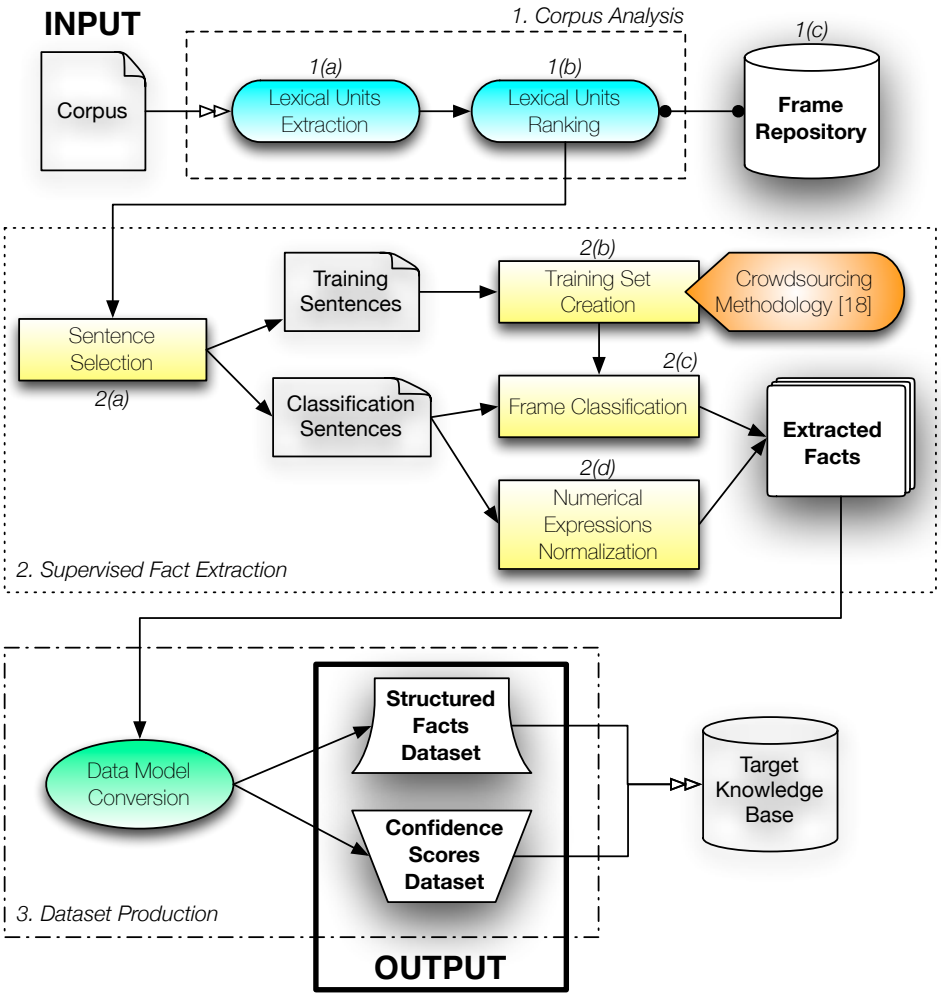

Figure 4.2: High level overview of the *Relation Extractor* system

4.3 System Description

The implementation workflow is intended as follows, depicted in Figure 4.2, and applied to the use case in Italian language:

1. *Corpus Analysis*

   (a) **Lexical Units (LUs) Extraction** via text tokenization, lemmatization, and part-of-speech (POS) tagging. LUs serve as frame triggers;

   (b) **LUs Ranking** through lexicographical and statistical analysis of the input corpus. The selection of top-N meaningful LUs is produced via a combination of term weighting measures (i.e., TF-IDF) and purely statistical ones (i.e., standard deviation);

   (c) each selected LU will trigger one or more frames together with their FEs, depending on the definitions contained in a given **frame repository**. The repository also holds the input labels for two automatic classifiers (the former handling FEs, the latter frames) based on Support Vector Machines (SVM).

2. *Supervised Relation Extraction*

   (a) **Sentence Selection**: two sets of sentences are gathered upon the candidate LUs, one for training examples and the other for the actual classification;

   (b) **Training Set Creation**: construction of a fully annotated training set via crowdsourcing;

   (c) **Frame Classification**: massive frame and FEs extraction on the input corpus seed sentences, via the classifiers trained with the result of the previous step.

3. *Dataset Production*: structuring the extraction results to fit the target KB (i.e., DBpedia) **data model** (i.e., RDF). A frame would map to a property, while participants would either map to subjects or to objects, depending on their role.

We proceed with a simplification of the original Frame Semantics theory with respect to two aspects: (a) LUs may be evoked by additional POS (e.g., nouns), but we focus on verbs, since we assume that they are more likely to trigger factual information; (b) depending on the frame repository, full lexical coverage may not be guaranteed (i.e., some LUs may not trigger any frames), but we expect that ours will, otherwise LU candidates would not generate any fact.

4.4 Corpus Analysis

Since Wikipedia also contains semi-structured data, such as formatting templates, tables, references, images, etc., a pre-processing step is required to obtain the raw text representation only. To achieve this, we leverage a third-party tool, namely the WikiExtractor.[11] From the entire Italian Wikipedia corpus, we slice the use case subset by querying the Italian DBpedia chapter[12] for the Wikipedia article IDs of relevant entities.

4.4.1 Lexical Units Extraction

Given the use case corpus, we first extract the complete set of verbs through a standard NLP pipeline: tokenization, lemmatization and POS tagging. POS information is required to identify verbs, while lemmas are needed to build the ranking. TreeTagger[13] is exploited to fulfill these tasks.

[11] https://github.com/attardi/wikiextractor
[12] http://it.dbpedia.org/sparql
[13] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

4.4.2 Lexical Units Selection

The unordered set of extracted verbs needs to undergo a further analysis, which aims at discovering the most representative verbs with respect to the corpus. As a matter of fact, lexicon (LUs) in text is typically distributed according to the Zipf's law,[14] where few highly occurring terms cater for a vast portion of the corpus. Of course, grammatical words (stopwords) are the top-occurring ones, although they do not bear any meaning, and must be filtered. We can then focus on the most frequent LUs and benefit from two advantages: first, we ensure a wide coverage of the corpus with few terms; second, we minimize the annotation cost. To achieve this, we need to frame the selection as a ranking problem, where we catch a frequency signal in order to calculate a score for each LU. It is clear that processing the long tail of lowly occurring LUs will be very expensive and not particularly fruitful.

Two measures are leveraged to generate a score for each verb lemma. We first compute the term frequency–inverse document frequency (TF-IDF) of each verb lexicalization t belonging to the set of occurring tokens T over each document d in the corpus C: this weighting measure $\alpha_{t,d}$ is intended to capture the *lexicographical* relevance of a given verb, namely how important it is with respect to other terms in the whole corpus. Then, we determine the standard deviation value out of the TF-IDF scores set A_t: this *statistical* measure β_t is meant to catch heterogeneously distributed verbs, in the sense that the higher the standard deviation is, the more variably the verb is used, thus helping to understand its overall usage signal over the corpus. Ultimately, we produce the final score s and assign it to a verb lemma by averaging all its lexicalizations scores B. To clarify how the two measures

[14] https://en.wikipedia.org/wiki/Zipf%27s_law

are combined, we formalize the LU selection problem as follows.

$$\forall t \in T, \forall d \in C \text{ let } \alpha_{t,d} = tfidf(t,d);$$

$$A_t = \bigcup_{d \in C} \{\alpha_{t,d}\}; \quad \beta_t = stdev(A_t);$$

$$B = \bigcup_{t \in T} \{\beta_t\}; \quad s = avg(B)$$

The ranking is publicly available in the code repository.[15] The top-N lemmas serve as candidate LUs, each evoking one or more frames according to the definitions of a given frame repository.

4.5 Use Case Frame Repository

Among the top 50 LUs that emerged from the corpus analysis phase, we manually selected a subset of 5 items to facilitate the full implementation of our pipeline. Once the approach has been tested and evaluated, it can scale up to the whole ranking (cf. Section 4.11 for more observations). The selected LUs comply with two criteria: first, they are picked from both the best and the worst ranked ones, with the purpose of assessing the validity of the corpus analysis as a whole; second, they fit the use case domain, instead of being generic. Consequently, we proceed with the following LUs: **esordire** (to start out), **giocare** (to play), **perdere** (to lose), **rimanere** (to stay, remain), and **vincere** (to win).

The next step consists of finding a language resource (i.e., frame repository) to suitably represent the use case domain. Given a resource, we first need to define a relevant subset, then verify that both its frame and FEs definitions are a relevant fit. After an investigation of FrameNet and KICKTIONARY [112], we notice that:

[15] https://github.com/dbpedia/fact-extractor/blob/master/resources/stdevs-by-lemma.json

- to the best of our knowledge, no suitable domain-specific Italian FrameNet or Kicktionary are publicly available, in the sense that neither LU sets nor annotated sentences for the Italian language match our purposes;
- FrameNet is too coarse-grained to encode our domain knowledge. For instance, the FINISH_COMPETITION frame may seem a relevant candidate at a first glimpse, but does not make the distinction between a victory and a defeat (as it can be triggered by both `to win` and `to lose` LUs), thus rather fitting as a super-frame (but no sub-frames exist);
- Kicktionary is too specific, since it is built to model the speech transcriptions of football matches. While it indeed contains some in-scope frames such as VICTORY (evoked by `to win`), most LUs are linked to frames that are not likely to appear in our input corpus, e.g., `to play` with PASS (occurring in sentences like `Ronaldinho played the ball in for Deco`).

Therefore, we adopted a custom frame repository, maximizing the reuse of the available ones as much as possible, thus serving as a hybrid between FrameNet and Kicktionary. Moreover, we tried to provide a challenging model for the classification task, prioritizing FEs overlap among frames and LU ambiguity (i.e., focusing on very fine-grained semantics with subtle sense differences). We believe this does not only apply to machines, but also to humans: we can view it as a stress test both for the machine learning and the crowdsourcing parts. A total of 6 frames and 15 FEs are modeled with Italian labels as follows:

- ATTIVITÀ (activity), FEs AGENTE (agent), COMPETIZIONE (competition), DURATA (duration), LUOGO (place), SQUADRA (team), TEMPO (time). Evoked by `esordire` (to start out), `giocare` (to play), `rimanere`

(to stay, remain), as in `Roberto Baggio played with Juventus in Serie A between 1990 and 1995`. Frame label translated from FrameNet ACTIVITY, FEs from a subset of FrameNet ACTIVITY;

- PARTITA (match), FEs SQUADRA_1 (team 1), SQUADRA_2 (team 2), COMPETIZIONE, LUOGO, TEMPO, PUNTEGGIO (score), CLASSIFICA (ranking). Evoked by `giocare`, `vincere` (to win), `perdere` (to lose), as in `Juventus played Milan at the UEFA cup final (2-0)`. Frame label translated from Kicktionary MATCH, FEs from a subset of FrameNet COMPETITION, LU shared by both;

- SCONFITTA (defeat), FEs PERDENTE, VINCITORE, COMPETIZIONE, LUOGO, TEMPO, PUNTEGGIO, CLASSIFICA. Sub-frame of PARTITA, evoked by `perdere`, as in `Milan lost 0-2 against Juventus at the UEFA cup final`. Frame label translated from Kicktionary DEFEAT, FEs from a subset of FrameNet BEAT_OPPONENT, LU from Kicktionary;

- STATO (status), FEs ENTITÀ (entity), STATO (status), DURATA, LUOGO, SQUADRA, TEMPO. Evoked by `rimanere`, as in `Roberto Baggio remained faithful to Juventus until 1995`. Custom frame and FEs derived from corpus evidence, to augment the `rimanere` LU ambiguity;

- TROFEO (trophy), FEs AGENTE, COMPETIZIONE, SQUADRA, PREMIO (prize), LUOGO, TEMPO, PUNTEGGIO, CLASSIFICA. Sub-frame of PARTITA, evoked by `vincere`, as in `Roberto Baggio won a UEFA cup with Juventus in 1992`. Custom frame label, FEs from a subset of FrameNet WIN_PRIZE, LU from FrameNet;

- VITTORIA (victory), FEs VINCITORE, PERDENTE, COMPETIZIONE, LUOGO, TEMPO, PUNTEGGIO, CLASSIFICA. Evoked by `vincere`, as

in `Juventus won 2-0 against Milan at the UEFA cup final`. Frame label translated from Kicktionary VICTORY, FEs from a subset of FrameNet BEAT_OPPONENT, LU from Kicktionary.

4.6 Supervised Relation Extraction

The first stage involves the creation of the training set: we leverage the crowdsourcing platform CROWDFLOWER[16] and a one-step frame annotation method, which we briefly illustrate in Section 4.6.2. The training set has a double outcome, as it will feed two classifiers: one will identify FEs, and the other is responsible for frames.

Both frame and FEs recognition are cast to a multi-class classification task: while the former can be related to text categorization, the latter should answer questions such as *"can this entity be this FE?"* or *"is this entity this FE in this context?"*. Such activity boils down to semantic role labeling (cf. [77] for an introduction), and usually requires a more fine-grained text analysis. Previous work in the area exploits deeper NLP layers, such as syntactic parsing (e.g., [79]). We alleviate this through Entity Linking (EL) techniques, which perform word sense disambiguation by linking relevant parts of a source sentence to URIs of a target KB. We leverage THE WIKI MACHINE as our EL approach (cf. Chapter 2, Section 2.1). EL results are part of the FE classifier feature set. We claim that EL enables the automatic addition of features based on existing entity attributes within the target KB (notably, the class of an entity, which represents its semantic type).

Given as input an unknown sentence, the full frame classification workflow involves the following tasks: tokenization, POS tagging, EL, FE classification, and frame classification.

[16] http://www.crowdflower.com/

4.6.1 Sentence Selection

The sentence selection procedure allows to harvest meaningful sentences from the input corpus, and to feed the classifier. Therefore, its outcome is two-fold: to build a representative training set and to extract relevant sentences for classification. We experimented multiple strategies as follows. They all share the same base constraint, i.e., each seed must contain a LU lexicalization.

- *Baseline*: the seed must be comprised in a given interval of length in words;
- *Sentence splitter*: the seed forms a complete sentence extracted with a sentence splitter. This strategy requires training data for the splitter;
- *Chunker grammar*: the seed must match a pattern expressed via a context-free chunker grammar. This strategy requires a POS tagger and engineering effort for defining the grammar (e.g., a noun phrase, followed by a verb phrase, followed by a noun phrase);
- *Syntactic*: the seed is extracted from a parse tree obtained through immediate constituent analysis, the idea being to split long and complex sentences into shorter ones. This strategy requires a suitable grammar and a parser;
- *Lexical*: the seed must match a pattern based on lexicalizations of candidate entities. This strategy requires querying a KB for instances of relevant classes (e.g., soccer-related ones as per the use case).

First, we note that all the strategies but the baseline necessitate an evident cost overhead in terms of language resources availability and engineering. Furthermore, given the soccer use case input corpus of $52,000$ articles circa, all strategies but the syntactic one dramatically reduce the number

of seeds, while the baseline performed an extraction with a .95 article/seed ratio (despite some noise). Compared to the sentence splitter strategy, the syntactic one brought an increase of roughly 4x in the number of seeds, at a cost of 375x in processing time, which we deemed not worth. These numbers arise from an experiment carried out for Wikidata, with a larger corpus composed of 500,000 documents circa from heterogeneous Web sources (cf. Section 4.11.3).

Consequently, we decided to leverage the baseline for the sake of simplicity and for the compliance to our contribution claims. We set the interval to $5 < w < 25$, where w is the number of words. The selection of relatively concise sentences is motivated by empirical and conceptual reasons:

(a) it is known that crowdsourced NLP tasks should be as simple as possible [116]. Hence, it is vital to maximize the accessibility, otherwise the job would be too confusing and frustrating, with a consistent impact in quality and execution time;

(b) frame annotation is a particularly complex task [7], even for expert linguists. Therefore, the inter-annotator agreement is expected to be fairly low. Compact sentences minimize disagreement, as corroborated by the average score we obtained in the gold standard (cf. Section 4.10.1, Table 4.3 and 4.4);

(c) since we aim at populating a KB, we prioritize precise statements instead of recall, for the sake of data quality. As a result, we focus on atomic factual information to reduce the risk of noise;

(d) on the light of the above points, Entity Linking acts as a surrogate of syntactic parsing, thus complying with our initial claim.

We still foresee further investigation of the other strategies for scaling besides the use case. Specifically, we believe that the refinement of the

chunker grammar would be the most beneficial approach: POS tagging is already involved into the system architecture, thus allowing to concentrate the engineering costs on the grammar only.

4.6.2 Training Set Creation

We apply a one-step, bottom-up approach to let the crowd perform a full frame annotation over a set of training sentences. In Frame Semantics, lexical ambiguity is represented by the number of frames that a LU may trigger. For instance, `vincere` (to win) conveys TROFEO (trophy) and VITTORIA (victory), thus having an ambiguity value of 2. The idea is to directly elicit the detection of *core* FEs, which are the essential items allowing to discriminate between frames. In this way, we are able to both annotate the FEs and let the correct frame emerge, thus also disambiguating the LU. The objective is achieved as follows: given a sentence s holding a LU with frame set F and set cardinality (i.e., ambiguity value) n, we solicit n annotations of s, and associate each one to the core FEs of each frame $f \in F$. We allow workers to select the `None` answer, and infer the correct frame based on the amount of `None`.

The training set is randomly sampled from the input corpus and contains $3,055$ items. The outcome is the same amount of frame examples and $55,385$ FE examples. The task is sent to the CrowdFlower platform.

Crowdsourcing Caveats

Swindles represent a widespread pitfall of crowdsourcing services: workers are usually rewarded a very low monetary amount (i.e., a few cents) for jobs that can be finalized with a single mouse click. Therefore, the results are likely to be excessively contaminated by random answers. CrowdFlower

Figure 4.3: Worker interface example

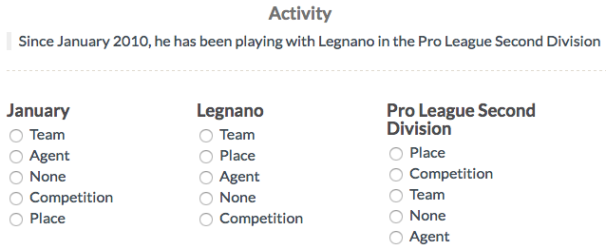

Figure 4.4: Worker interface example translated in English

tackles the problem via *test questions*,[17] namely data units which are pre-marked with the correct response. If a worker fails to meet a given minimum accuracy threshold,[18] he or she will be labeled as *untrusted* and his or her contribution will be automatically rejected.

Task Design

We ask the crowd to (a) read the given sentence, (b) focus on the "topic" (i.e., the potential frame that disambiguates the LU) written above it, and (c) assign the correct "label" (i.e., the FE) to each "word" (i.e., unigram) or "group of words" (i.e., n-grams) from the multiple choices provided below each n-gram. Figure 4.3 displays the front-end interface of a sample sentence, with Figure 4.4 being its English translation.

[17] https://success.crowdflower.com/hc/en-us/articles/202703305-Getting-Started-Glossary-of-Terms#test_question

[18] https://success.crowdflower.com/hc/en-us/articles/202702975-Job-Settings-Guide-To-Test-Question-Settings-Quality-Control

During the preparation phase of the task input data, the main challenge is to automatically provide the crowd with relevant candidate FE text chunks, while minimizing the production of noisy ones. To tackle this, we experimented with the following chunking strategies:

- third-party full-stack NLP pipeline, namely TextPro [98] for Italian, by extracting nominal chunks with the ChunkPro module;[19]
- custom noun phrase chunker via a context-free grammar;
- EL surface forms;

We surprisingly observed that the full-stack pipeline outputs a large amount of noisy chunks, besides being the slowest strategy. On the other hand, the custom chunker was the fastest one, but still too noisy to be crowdsourced. EL resulted in the best trade-off, and we adopted it for the final task.

The task parameters are as follows:

- we set 3 judgments per sentence to enable the computation of an agreement based on majority vote;
- the pay sums to 5 $ cents per page, where one page contains 5 sentences;
- we limit the task to Italian native speakers only by targeting the Italian country and setting the required language skills to Italian;
- the minimum worker accuracy is set to 70% in quiz mode (i.e., the warm-up phase where workers are only shown gold units and are recruited according to their accuracy) and relaxed to 65% in work mode (i.e., the actual annotation phase) to avoid extra cost in terms of time and expenses to collect judgments;

[19] http://textpro.fbk.eu/

Table 4.2: Training set crowdsourcing task outcomes. Cf. Section 4.6.2 for explanations of CrowdFlower-specific terms

Sentences	3,111
Test questions	56
Trusted judgments	9,198
Untrusted judgments	972
Total cost	152.46 $

- on account of a personal calibration, the minimum time per page threshold is set to 30 seconds, which allows to automatically discard a contributor when triggered;

- we set the maximum number of judgments per contributor to 280, in order to prevent each contributor from answering more than once on a given sentence, while avoiding to remove proficient contributors from the task.

The outcomes are resumed in Table 4.2.

Finally, the crowdsourced annotation results are processed and translated into a suitable format to serve as input training data for the classifier.

4.6.3 Frame Classification: Features

We train our classifiers with the following linguistic features, in the form of bag-of-features vectors:

1. *both classifiers*: for each input word token, both the token itself (bag of terms) and the lemma (bag of lemmas);

2. *FE classifier*: contextual sliding window of width 5 (i.e., 5-gram, for each token, consider the 2 previous and the 2 following ones);

3. *frame classifier*: we implement our bottom-up frame annotation approach, thus including the set of FE labels (bag of roles) to help this classifier induce the frame;

4. *gazetteer*: defined as a map of key-value pairs, where each key is a feature and its value is a list of n-grams, we automatically build a wide-coverage gazetteer with relevant DBpedia ontology (DBPO) classes as keys (e.g., `SoccerClub`) and instances as values (e.g., `Juventus`), by way of a query to the target KB.

4.7 Numerical Expressions Normalization

During the pilot crowdsourcing annotation experiments, we noticed a low agreement on numerical FEs. This is likely to stem from the FE labels interpretation: workers got particularly confused by TIME and DURATION, which explains the low agreement. Moreover, asking the crowd to label such frequently occurring FEs would represent a considerable overhead, resulting in a higher temporal cost (i.e., more annotations per sentence) and lower overall annotation accuracy. Hence, we opted for the implementation of a rule-based system to detect and normalize numerical expressions. The normalization process takes as input a numerical expression such as a date, a duration, or a score, and outputs a transformation into a standard format suitable for later inclusion into the target KB.

The task is not formulated as a classification one, but we argue it is relevant for the completeness of the extracted facts: rather, it is carried out via matching and transformation rule pairs. Given for instance the input expression `tra il 1920 e il 1925` (between 1920 and 1925), our normalizer first matches it through a regular expression rule, then applies a transformation rule complying to the XML Schema Datatypes[20] (typically

[20] http://www.w3.org/TR/xmlschema-2/

dates and times) standard, and finally produces the following output:[21]

```
duration: "P5Y"^^xsd:duration
start: "1920"^^xsd:gYear
end: "1925"^^xsd:gYear
```

All rule pairs are defined with the programming language-agnostic YAML[22] syntax. The pair for the above example is as follows. Regular Expression:

```
tra il (?P<y1>\ d{{2,4}}) e il (?P<y2>\ d{{2,4}})
```

Transformation:

```
{
  'duration':
'"P{}Y"^^<{}>'.format(
int(match.group('y2')) - int(match.group('y1')),
schema['duration']
),
  'start':
'"{}"^^<{}>'.format(
abs_year(match.group('y1')), schema['year']
),
  'end':
'"{}"^^<{}>'.format(
abs_year(match.group('y2')), schema['year']
)
}
```

In total, we have identified 21 rules, which are publicly available for consultation.[23]

4.8 Dataset Production

The integration of the extraction results into DBpedia requires their conversion to a suitable data model, i.e., RDF. Frames intrinsically bear N-ary

[21] We use the `xsd` prefix as a short form for the full URI `http://www.w3.org/2001/XMLSchema#`

[22] http://www.yaml.org/spec/1.2/spec.html

[23] https://github.com/dbpedia/fact-extractor/blob/master/date_normalizer/regexes.yml

relations through FEs, while RDF naturally represents binary relations. Hence, we need a method to express FEs relations in RDF, namely *reification*. This can be achieved in multiple ways:

- standard reification;[24]
- N-ary relations,[25] an application of Neo-Davidsonian representations [109, 108], with similar efforts [42, 62];
- named graphs.[26]

A recent overview [61] highlighted that all the mentioned strategies are similar with respect to query performance. Given as input n frames and m FEs, we argue that:

- standard reification is too verbose, since it would require $3(n + m)$ triples;
- applying Pattern 1 of the aforementioned W3C Working Group note to N-ary relations would allow us to build $n + m$ triples;
- named graphs can be used to encode provenance or context metadata, e.g., the article URI from where a fact was extracted. In our case however, the fourth element of the quad would be the frame (which represents the context), thus boiling down to minting $n + m$ quads instead of triples;

We opted for the less verbose strategy, namely N-ary relations. Given the running example sentence `In Euro 1992, Germany reached the final, but lost 0–2 to Denmark`, classified as a DEFEAT frame and embedding the FEs WINNER, LOSER, COMPETITION, SCORE, we generate RDF as per the following Turtle serialization:

[24] http://www.w3.org/TR/2004/REC-rdf-primer-20040210/#reification
[25] http://www.w3.org/TR/swbp-n-aryRelations/
[26] http://www.w3.org/TR/rdf11-concepts/

```
:Germany :defeat :Defeat_01 .
:Defeat_01
   :winner :Denmark ;
   :loser :Germany ;
   :competition :Euro_1992 ;
   :score "0-2" .
```

We add an extra instance type triple to assign an ontology class to the reified frame, as well as a provenance triple to indicate the original sentence:

```
:Defeat_01
   a :Defeat ;
   :extractedFrom "In Euro 1992,
      Germany reached the final,
      but lost 0{2 to Denmark"@it .
```

In this way, the generated statements amount to $n + m + 2$.

It is not trivial to decide on the subject of the main frame statement, since not all frames are meant to have exactly one core FE that would serve as a plausible logical subject candidate: most have many, e.g., FINISH_COMPETITION has COMPETITION, COMPETITOR and OPPONENT as core FEs in FrameNet. Therefore, we tackle this as per the following assumption: given the encyclopedic nature of our input corpus, both the logical and the topical subjects correspond in each document. Hence, each candidate sentence inherits the document subject. We acknowledge that such assumption strongly depends on the corpus: it applies to entity-centric documents, but will not perform well for general-purpose ones such as news articles. However, we believe it is still a valid in-scope solution fitting our scenario.

Confidence Scores

Besides the fact datasets, we also keep track of confidence scores and generate additional datasets accordingly. Therefore, it is possible to filter facts that are not considered as confident by setting a suitable threshold. When processing a sentence, our pipeline outputs two different scores for each FE, stemming from the entity linker and the supervised classifier. We merge both signals by calculating the F-score between them, as if they were representing precision and recall, in a fashion similar to the standard classification metrics. The global fact score can be then produced via an aggregation of the single FE scores in multiple ways, namely: (a) arithmetic mean; (b) weighted mean based on core FEs (i.e., they have a higher weight than extra ones); (c) harmonic mean, weighted on core FEs as well.

The reader may refer to Section 4.11.5 for a distributional analysis of these scores over the output dataset.

4.9 Baseline Classifier

To enable a performance evaluation comparison with the supervised method, we developed a rule-based algorithm that handles the full frame and FEs annotation. The main intuition is to map FEs defined in the frame repository to ontology classes of the target KB: such mapping serves as a set of rule pairs $(FE, class)$, e.g., (WINNER, `SoccerClub`). In the FrameNet terminology, this is homologous to the assignment of *semantic types* to FEs: for instance, in the ACTIVITY frame, the AGENT is typed with the generic class `Sentient`. The idea would allow the implementation of the bottom-up one-step annotation flow described in [50]: to achieve this, we run EL over the input sentences and check whether the attached ontology class metadata appear in the frame repository, thus fulfilling the FE classification task.

Besides that, we exploit the notion of core FEs: this would cater for the

frame disambiguation part. Since a frame may contain at least one core FE, we proceed with a *relaxed* assignment, namely we set the frame if a given input sentence contains at least one entity whose ontology class maps to a core FE of that frame. The implementation workflow is illustrated in Algorithm 1: it takes as input the set S of sentences, the frame repository F embedding frame and FEs labels, core/non-core annotations and rule pairs, and the set L of trigger LU tokens.

It is expected that the relaxed assignment strategy will not handle the overlap of FEs across competing frames that are evoked by a single LU. Therefore, if at least one core FE is detected in multiple frames, the baseline makes a random assignment for the frame. Furthermore, the method is not able to perform FE classification in case different FEs share the ontology class (e.g., both WINNER and LOSER map to `SoccerClub`): we opt for a FE random guess as well.

4.10 Evaluation

We assess our main research contributions through the analysis of the following aspects:

- Classification performance;
- T-Box property coverage extension;
- A-Box statements addition;
- final fact correctness.

4.10.1 Classification Performance

We assess the overall performance of the baseline and the supervised systems over a gold standard dataset. We randomly sampled 500 sentences

containing at least one occurrence of our use case LU set from the input corpus. We first outsourced the annotation to the crowd as per the training set construction and the results were further manually validated twice by the authors. CrowdFlower provides a report including an agreement score for each answer, computed via majority vote weighted by worker trust: we calculated the average among the whole evaluation set, obtaining a value of .916.

With respect to the FEs classification task, we proceed with 2 evaluation settings, depending on how FE text chunks are treated, namely:

- **lenient**, where the predicted ones at least *partially* match the expected ones;

- **strict**, where the predicted ones must *perfectly* match the expected ones.

Table 4.3 illustrates the outcomes. FE measures are computed as follows: (1) a true positive is triggered if the predicted label is correct and the predicted text chunk matches the expected one (according to each setting); chunks that should not be labeled are marked with a "O" and (2) not counted as true positives if the predicted ones are correct, but (3) indeed counted as false positives in the opposite case. The high frequency of "O" occurrences (circa 80% of the total) in the gold standard actually penalizes the system, thus providing a more challenging evaluation playground.

On the other hand, the frame classification task does not need to undergo chunk assessment, since it copes with the whole input sentence. Therefore, the lenient and strict settings are not applicable, and we proceed with a standard evaluation. The results are reported in Table 4.4.

Table 4.3: Frame Elements (FEs) classification performance evaluation over a gold standard of 500 random sentences from the Italian Wikipedia corpus. The average crowd agreement score on the gold standard amounts to .916

Approach	Lenient			Strict		
	P	**R**	**F1**	**P**	**R**	**F1**
Baseline	73.48	65.83	69.45	67.68	63.79	65.68
Supervised	83.33	75.00	78.94	73.59	66.66	69.96

Supervised Classification Performance Breakdown

Figure 4.5 and Figure 4.7 respectively display the FE and frame classification confusion matrices: they are normalized such that the sum of elements in the same row is 1. Since we highlight the cells through a color scale, the normalization is needed to avoid too similar color nuances that would originate from absolute results.

FEs. We observe that COMPETIZIONE is frequently mistaken for PREMIO and ENTITÀ, while rarely for TEMPO and DURATA, or just missed. On the other hand, TEMPO is mistaken for COMPETIZIONE: our hypothesis is that competition mentions, such as `World Cup 2014`, are disambiguated as a whole entity by the linker, since a specific target Wikipedia article exists. However, it overlaps with a temporal expression, thus confusing the classifier. AGENTE is often mistaken for ENTITÀ, due to their equivalent

Table 4.4: Frame classification performance evaluation over a gold standard of 500 random sentences from the Italian Wikipedia corpus. The average crowd agreement score on the gold standard amounts to .916

Approach	P	R	F1
Baseline	74.25	62.50	67.87
Supervised	84.35	82.86	83.60

Table 4.5: Lexicographical analysis of the Italian Wikipedia soccer player sub-corpus

Stems (frequency %)	**Candida**
gioc (47), partit (39), campionat (34), stagion (36), presen (30), disput (20), serie (14), nazional (13), titolar (13), competizion (5), scend (5), torne (5)	Compet
pass (24), trasfer (19), prest (15), contratt (11)	Activit
termin (12), contratt, ced (10), lasc (6), vend (2)	Activit
gioc, disput (20), scend	Finish_c
campionat, stagion, serie, nazional, competizion, torne	Finish_c
vins/vinc (18), pers/perd (11), sconfi (8)	Beat_o
vins/vinc, conquis (8), otten (7), raggiun (6), aggiud (2)	Win_pri

semantic type, which is always a person.

Frames. We note that Attività is often mistaken for Stato or not classified at all: in fact, the difference between these two frames is quite subtle with respect to their sense. The former is more generic and could also be labeled as Career: if we viewed it in a frame hierarchy, it would serve as a super-frame of the latter. The latter instead encodes the development modality of a soccer player's career, e.g., when he remains unbound from some team due to contracting issues. Hence, we may conclude that distinguishing between these frames is a challenge even for humans.

Furthermore, frames with no FEs are classified as "O", thus considered wrong despite the correct prediction. Vittoria is almost never mistaken for Trofeo: this is positively surprising, since the FE Competizione (frame Vittoria) is often mistaken for Premio (frame Trofeo), but those FEs do not seem to affect the frame classification. Again, such FE distinction must take into account a delicate sense nuance, which is hard for humans as well.

Figure 4.6 and Figure 4.8 respectively plot the FE and frame classification performance, broken down to each label.

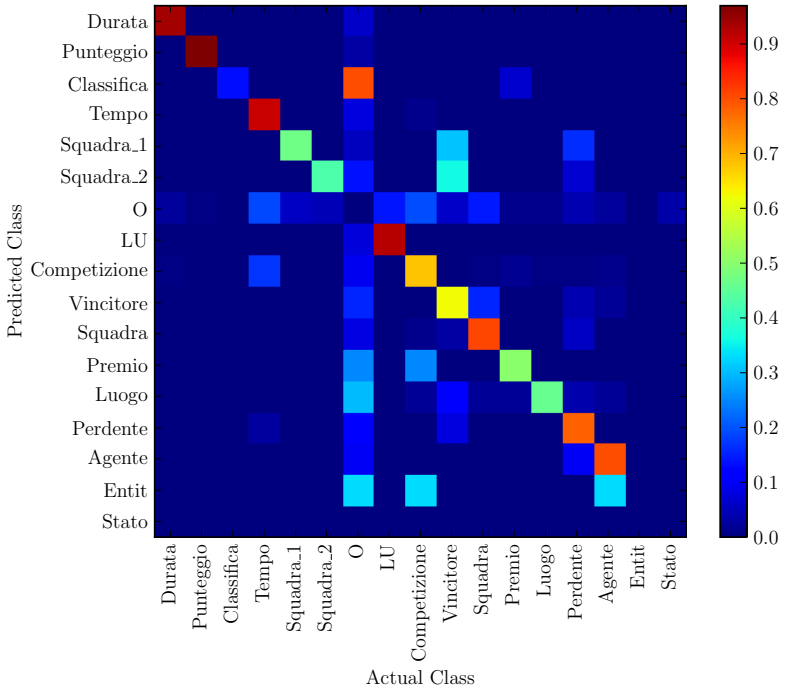

Figure 4.5: Supervised FE classification normalized confusion matrix, lenient evaluation setting. The color scale corresponds to the ratio of predicted versus actual classes. Normalization means that the sum of elements in the same row must be 1.0

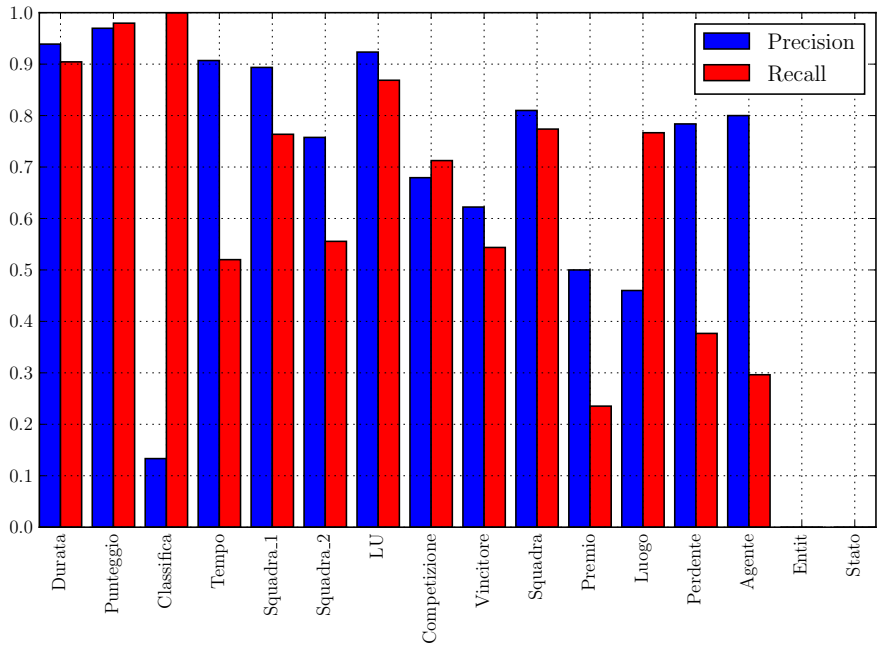

Figure 4.6: Supervised FE classification precision and recall breakdown, lenient evaluation setting

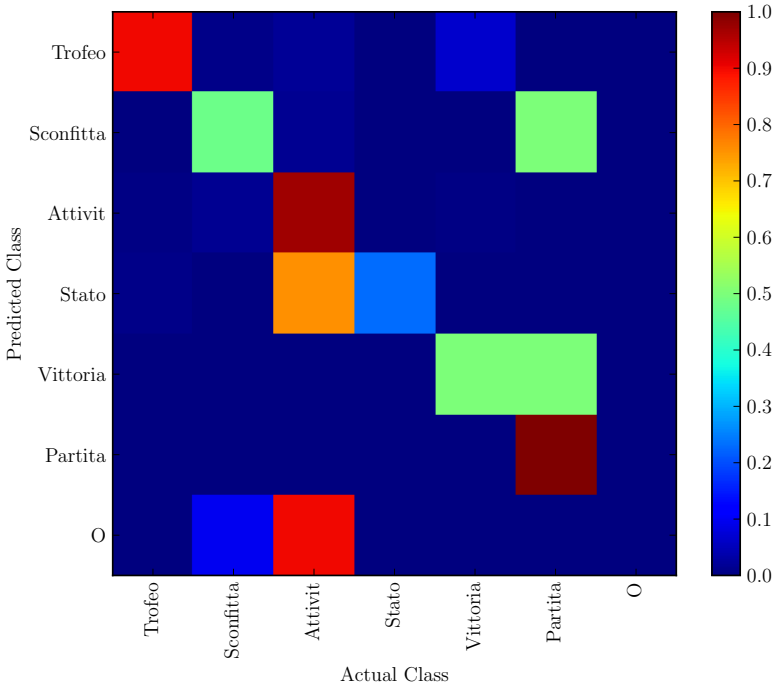

Figure 4.7: Supervised frame classification normalized confusion matrix. The color scale corresponds to the ratio of predicted versus actual classes. Normalization means that the sum of elements in the same row must be 1.0

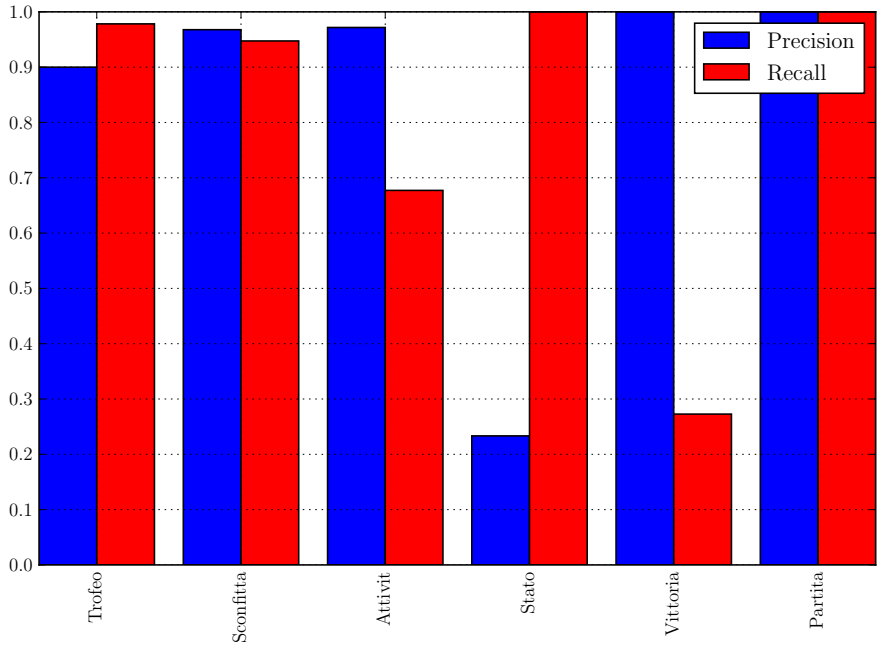

Figure 4.8: Supervised frame classification precision and recall breakdown

4.10.2 T-Box Enrichment

One of our main objectives is to extend the target KB ontology with new properties on existing classes. We focus on the use case and argue that our approach will have a remarkable impact if we manage to identify non-existing properties. This would serve as a proof of concept which can ideally scale up to all kinds of input. In order to assess such potential impact in discovering new relations, we need to address the following question: *"which extractable relations are not already mapped in DBPO or do not even exist in the raw infobox properties datasets?"*. Table 4.5 illustrates an empirical lexicographical study gathered from the Italian Wikipedia soccer player sub-corpus (circa 52,000 articles). It contains occurrence frequency percentages of word stems (in descending order) that are likely to trigger domain-relevant frames, thus providing a rough overview of the extraction potential.

The corpus analysis phase (cf. Section 4.4) yielded a ranking of LUs evoking the frames ACTIVITY, DEFEAT, MATCH, TROPHY, STATUS, and VICTORY: these frames would serve as ontology property candidates, together with their embedded FEs. DBPO already has most of the classes that are needed to represent the main entities involved in the use case: `SoccerPlayer`, `SoccerClub`, `SoccerManager`, `SoccerLeague`, `SoccerTournament`, `SoccerClubSeason`, `SoccerLeagueSeason`, although some of them lack an exhaustive description (cf. `SoccerClubSeason`[27] and `SoccerLeagueSeason`).[28]

For each of the 7 aforementioned DBPO classes, we computed the amount and frequency of ontology and raw infobox properties by querying the Italian DBpedia endpoint. Results (in ascending order of frequency) are publicly

[27] http://mappings.dbpedia.org/server/ontology/classes/SoccerClubSeason
[28] http://mappings.dbpedia.org/server/ontology/classes/SoccerLeagueSeason

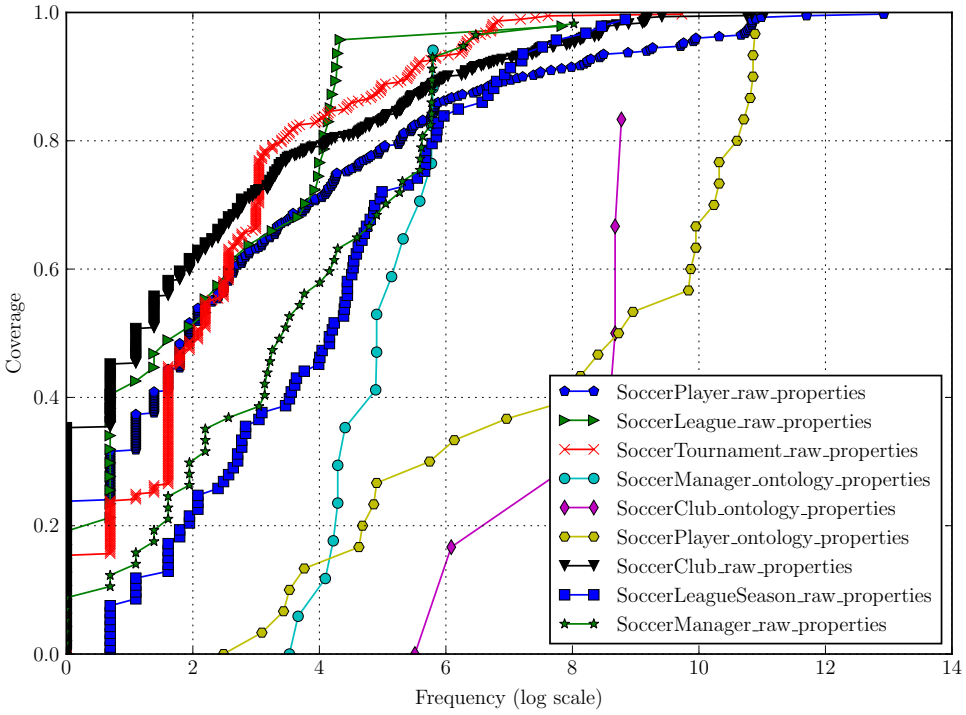

Figure 4.9: Italian DBpedia soccer property statistics

available,[29] and Figure 4.9 illustrates their distribution. The horizontal axis stands for the normalized (log scale) frequency, encoding the current usage of properties in the target KB; the vertical axis represents the ratio (which we call coverage) between the position of the property in the ordered result set of the query and the total amount of distinct properties (i.e., the size of the result set). Properties with a null frequency are ignored.

First, we observe a lack of ontology property usage in 4 out of 7 DBPO classes, probably due to missing mappings between Wikipedia template attributes and DBPO. On the other hand, the ontology properties have a more homogenous distribution compared to the raw ones: this serves as an expected proof of concept, since the main purpose of DBPO and the ontology mappings is to merge heterogenous and multilingual Wikipedia template attributes into a unique representation. On average, most raw

[29] http://it.dbpedia.org/downloads/fact-extraction/soccer_statistics/

properties are concentrated below coverage and frequency threshold values of 0.8 and 4 respectively: this means that roughly 80% are rarely used, and the log scale further highlights the evidence. While ontology properties are better distributed, most still do not reach a high coverage/frequency trade-off, except for `SoccerPlayer`, which benefits from both rich data (cf. Section 4.2) and mappings.[30]

On the light of the two analyses discussed above, it is clear that our approach would result in a larger variety and finer granularity of facts than those encoded into Wikipedia infoboxes and DBPO classes. Moreover, we believe the lack of dependence on infoboxes would enable more flexibility for future generalization to sources beyond Wikipedia.

Subsequent to the use case implementation, we manually identified the following mappings from frames and FEs to DBPO properties:

- Frames: (ACTIVITY, `careerStation`), (AWARD, `award`), (STATUS, `playerStatus`);
- FEs: (TEAM, `team`), (SCORE, `score`), (DURATION, [`duration`, `startYear`, `endYear`]).

Our system would undeniably benefit from a property matching facility to discover more potential mappings, although a research contribution in ontology alignment is out of scope for this work. In conclusion, we claim that 3 out 6 frames and 12 out of 15 FEs represent novel T-Box properties.

4.10.3 A-Box Population

Our methodology enables a simultaneous T-Box and A-Box augmentation: while frames and FEs serve as T-Box properties, the extracted facts feed the A-Box part. Out of 49,063 input sentences, we generated a total of 213,479

[30] http://mappings.dbpedia.org/index.php/Mapping_it:Sportivo

Table 4.6: Relative A-Box population gain compared to pre-existing T-Box property assertions in the Italian DBpedia chapter

Property	Dataset	Assertions (#)	Gain (%)
careerStation	DBpedia	2,073	N.A.
	Baseline all	20,430	89.8
	Supervised all	26,316	92.12
award	DBpedia	7,755	N.A.
	Baseline all	4,953	-56.57
	Supervised all	10,433	25.66
playerStatus	DBpedia	0	N.A.
	Baseline all	0	0
	Supervised all	26	100

and 216,451 triples (i.e., with a 4.35 and 4.41 ratio per sentence) from the supervised and the baseline classifiers respectively. 52% and 55% circa are considered *confident*, namely facts with confidence scores (cf. Section 4.8) above the dataset average threshold.

To assess the domain coverage gain, we can exploit two signals: (a) the amount of produced novel data with respect to pre-existing T-Box properties and (b) the overlap with already extracted assertions, regardless of their origin (i.e., whether they stem from the raw infobox or the ontology-based extractors). Given the same Italian Wikipedia dump input dating 21 January 2015, we ran both the baseline and the supervised relation extraction, as well as the DBpedia extraction framework to produce an Italian DBpedia chapter release, thus enabling the coverage comparison.

Table 4.6 describes the analysis of signal (a) over the 3 frames that are mapped to DBPO properties. For each property and dataset, we computed the amount of available assertions and reported the gain relative to the relation extraction datasets. Although we considered the whole Italian DBpedia KB in these calculations, we observe that it has a generally low

coverage with respect to the analyzed properties, probably due to missing ontology mappings. For instance, the amount of assertions is always zero if we analyze the use case subset only, as no specific relevant mappings (e.g., `Carriera_sportivo`[31] to `careerStation`) currently exist. We view this as a major achievement, since our automatic approach also serves as a substitute for the manual mapping procedure.

Table 4.7 shows the results for signal (b). To obtain them, we proceeded as follows.

1. slice the use case DBpedia subset;
2. gather the subject-object patterns from all datasets. Properties are not included, as they are not comparable;
3. compute the patterns overlap between DBpedia and each of the relation extraction datasets (including the confident subsets);
4. compute the gain in terms of novel assertions relative to the relation extraction datasets.

The A-Box enrichment is clearly visible from the results, given the low overlap and high gain in all approaches, despite the rather large size of the DBpedia use case subset, namely $6,167,678$ assertions.

4.10.4 Final Fact Correctness

We estimate the overall correctness of the generated statements via an empirical evaluation over a sample of the output dataset. In this way, we are able to conduct a more comprehensive error analysis, thus isolating the performance of those components that play a key role in the extraction of facts: the Frame Semantics classifier, the numerical expression normalizer, and an external yet crucial element, i.e., the entity linker.

[31] https://it.wikipedia.org/wiki/Template:Carriera_sportivo

To achieve so, we randomly selected 10 instances for each frame from the supervised dataset and retrieve all the related triples. We excluded instance type triples (cf. Section 4.8), which are directly derived from the reified frame ones. Then, we manually assessed the validity of each triple element and assigned it to the component responsible for its generation. Finally, we checked the correctness of the whole triple.

More formally, given the evaluation set of triples E, the frame predicates set F, the non-numerical FE predicates set $\bar{N}$, and the numerical FE predicates set N (cf. Section 4.5), relevant triple elements are added to the classifier C, the normalizer N, the linker L, and to the set of all facts A as follows.

$$E \subseteq S \times P \times O;$$

$$P = F \cup \bar{N} \cup N; \qquad F \cap \bar{N} \cap N = \emptyset;$$

$$p_c \in F \cup \bar{N}; \qquad p_n \in N;$$

$$O = O_c \cup O_n; \qquad O_c \cap O_n = \emptyset;$$

$$o_c \in O_c; \qquad o_n \in O_n;$$

$$\forall (s, p, o) \in E \text{ let}$$

$$C \leftarrow C \cup \{(p_c, o_c)\}; \quad N \leftarrow N \cup \{(p_n, o_n)\};$$

$$L \leftarrow L \cup \{o_c\}; \qquad A \leftarrow A \cup \{(s, p, o)\}$$

Table 4.8 summarizes the outcomes.

Table 4.7: Overlap with pre-existing assertions in the Italian DBpedia chapter and relative gain in A-Box population

Dataset	**Overlap (#)**	**Gain (%)**
Baseline all	3,341	98.2
Supervised all	4,546	97.4
Baseline confident	2,387	97.6
Supervised confident	2,841	96.8

Table 4.8: Fact correctness evaluation over 132 triples randomly sampled from the supervised output dataset. Results indicate the ratio of correct data for the whole fact (**All**) and for triple elements produced by the main components of the system, namely: **Classifier**, as per Figure 4.2, part 2(c), and Section 4.6; **Normalizer**, as per Figure 4.2, part 2(d), and Section 4.7; **Linker**, external component, as per Section 4.6.

Classifier	Normalizer	Linker	All
.763	.820	.430	.727

Discussion

First, we observe that all the results but the linker are in line with our classification performance assessments detailed in Section 4.10.1. Accordingly, we notice that most of the errors involve the linker. More specifically, we summarize below an informal error analysis:

- generic dates appearing without years (as in `the 13th of August`) are resolved to their Wikipedia page.[32] These occurrences are then wrongly classified as COMPETIZIONE, consistently with what we remarked in Section 4.10.1;
- country names, e.g., `Sweden` are often linked to their national soccer team or to the major national soccer competition. This seems to mislead the classifier, which assigns a wrong role to the entity, instead of PLACE;
- the generic adjective `Nazionale` (national) is always linked to the Italian national soccer team, even though the sentence often contains enough elements to understand the correct country;
- some yearly intervals, e.g., `2010-2011` are linked to the corresponding season of the major Italian national soccer competition.

[32] https://en.wikipedia.org/wiki/August_13

Unfortunately, the linker tends to assign a fairly high confidence to these matches and so does the classifier, which assumes correct linking of entities. This leads to many assertions with undeserved high scores and underlines how important Entity Linking is in our pipeline.

4.11 Observations

We pinpoint and discuss here a list of notable aspects of this work.

4.11.1 LU Ambiguity

We acknowledge that the number of frames per LU in our use case repository may not be exhaustive to cover the potentially higher LU ambiguity. For instance, `giocare` (to play) may trigger an additional frame depending on the context (as in the sentence `to play as a defender`); `esordire` (to start out) may also trigger the frame PARTITA (match). Nevertheless, our one-step annotation approach is agnostic to the frame repository. Consequently, we expect that the LU ambiguity would not be an issue. Of course, the more a LU is ambiguous, the more expensive becomes the crowdsourcing job (cf. Section 4.6.2).

4.11.2 Manual Intervention Costs

Despite its low cost, we admit that crowdsourcing does not conceptually bypass the manual effort needed to create the training set: workers are indeed human annotators. However, we argue that the price can decrease even further by virtue of an automatic communication with the CrowdFlower API. This is already accomplished in the ongoing STREPHIT project, where we programmatically create jobs, post them, and pull their results. Hence, we may regard crowdsourcing as an activity that does not imply any direct

manual intervention by whoever runs the pipeline, if we exclude a minor quantity of test annotations, which are essential to reject cheaters.

Even though we recognize that the use case frame repository is hand-curated, we would like to emphasize that (a) it is intended as a test bed to assess the validity of our approach, and (b) its generalization should instead maximize the reuse of available resources. This is currently implemented in the StrepHit project, where we fully leverage FrameNet to look up relevant frames given a set of LUs.

4.11.3 NLP Pipeline Design

On account of our initial claim on the use of a shallow NLP machinery, we motivate below the choice of stopping to the grammatical layer. The decision essentially emanates from (1) the sentence selection phase, where we investigated several strategies, and (2) the construction of the crowdsourcing jobs, where we concurrently (2a) maximized the simplicity to smooth the way for the laymen workers, and (2b) automatically generated the candidate annotation chunks.

- *Chunking* is substituted by Entity Linking, as explored in Section 4.6.2;

- *Syntactic parsing* dramatically affects the computational costs, as shown in Table 4.9 and discussed in Section 4.6.1. Yet, we suppose that it could probably improve the performance in terms of recall. Given the KB population task, we still argue that precision should be made a priority, in order to produce high quality datasets;

- *Semantic Role Labeling* is not a requirement, since our system replaces this layer, as described in Section 4.6.

Table 4.9: Comparative results of the *Syntactic* sentence extraction strategy against the Sentence *Splitter* one, over a uniform sample of a corpus gathered from 53 Web sources, with estimates over the full corpus.

Strategy	# Documents	# Extracted	Cost
Splitter	7,929	13,846	1m 13s
Syntactic		41,205	6h 15m 49s
Splitter	504,189	899,159	1h 19m
Syntactic		2,675,853	16d 22h 45m 32s

4.11.4 Simultaneous T-Box and A-Box Augmentation

The Relation Extractor is conceived to extract factual information from text: as such, its primary output is a set of assertions that naturally feed the target KB A-Box. The T-Box enrichment is an intrinsic consequence of the A-Box one, since the latter provides evidence of new properties for the former. In other words, we adopt a data-driven method, which implies a bottom-up direction for populating the target KB. It is the duty of the corpus analysis module (Section 4.4) to understand the most meaningful relations between entities from the very bottom, i.e., the corpus. After that, the system proceeds upwards and translates the classification results into A-Box statements. These are already structured to ultimately carry the properties into the top layer of the KB, i.e., the T-Box.

4.11.5 Confidence Scores Distribution

Table 4.10 presents the cumulative (i.e., all FEs and frames aggregated) statistical distribution of confidence scores as observed in the gold standard. If we dig into single scores, we notice that the classifier usually outputs very high values for O and LU chunks, while average scores for other FEs range from .821 for Competition to .594 for Winner, down to .488 for

LOSER. On the other hand, EL scores have a relatively high average and a standard deviation of 0.273. In other words, the EL component is prone to set rather optimistic values, which are likely to have an impact on the global score.

Overall, due to the high presence of O chunks (circa 80% of the total), the EL and the classifier scores roughly match for each FE, and so do the final ones computed with the strategies introduced in Section 4.8. Assigning different weights to core and extra FEs has little impact on the global scores as well, varying their value by only 1 or 2% in both the weighted and the harmonic means. The arithmetic and weighted means yield the most optimistic global scores, averaging at .83 over the output dataset, while the harmonic mean settles at .75.

4.11.6 Scaling Up

Our approach has been tested on the Italian language, a specific domain, and with a small frame repository. Hence, we may consider the use case implementation as a monolingual closed-domain information extraction system. We outline below the points that need to be addressed for scaling up to multilingual open information extraction:

1. *Language*: training data availability for POS tagging and lemmatization. The LUs automatically extracted through the corpus analysis

Table 4.10: Cumulative confidence scores distribution over the gold standard

Type	**Min**	**Max**	**Avg**	**Stdev**
Classifier FEs	.181	.999	.945	.124
Classifier Frames	.412	.999	.954	.093
Links	.202	1.0	.697	.273
Global	.227	1.0	.838	.151

phase should be projected to a suitable frame repository;

2. *Domain*:

   - Baseline: mapping between FEs and target KB ontology classes;
   - Supervised:
     - financial resources for the crowdsourced training set construction, on average 4.79 $ cents per annotated sentence;
     - adapt the query to generate the gazetteer.

4.11.7 Crowdsourcing Generalization

With the Wikidata commitment in mind (Section 4.1), we aim at expanding our approach towards a corpus of non-Wikimedia Web sources and a broader domain. This entails the generalization of the crowdsourcing step. Overall, it has been proven that the laymen execute natural language tasks with reasonable performances [116]. Specifically, crowdsourcing Frame Semantics annotation has been recently shown to be feasible by [64]. Furthermore, [7] stressed the importance of eliciting non-expert annotators to avoid the high recruitment cost of linguistics experts. In [50], we further validated the results obtained by [64], and reported satisfactory accuracy as well. Finally, [25] proposed an approach to successfully scale up frame disambiguation.

On the light of the above references, we argue that the requirement can be indeed satisfied: as a proof of concept, we are working in this direction with StrepHit, where we have switched to a more extensive and heterogeneous input corpus. Here, we focus on a larger set L of LUs, thus $|L| \times n$ frames, where n is the average LU ambiguity. At the time of writing this paper, we are in the process of building the training set.

4.11.8 Miscellanea

First, if a sentence is not in the gold standard, the supervised classifier should discard it (abstention). Second, the baseline approach may contain rules that are more harmful than beneficial, depending on the target KB reliability: for instance, the `SportsEvent` DBPO class leads to wrongly typed instances, due to the misuse of the template by Wikipedia editors. Finally, both the input corpus and the target KB originate from a relatively small Wikipedia chapter (i.e., Italian, with 1.23 million articles) if compared to the largest one (i.e., English, with almost 5 million articles). Therefore, we recognize that the T-Box and A-Box evaluation results may be proportionally different if obtained with English data.

4.11.9 Technical Future Work

We report below a list of technical improvements left for planned implementation:

- LUs are handled as unigrams, but n-grams should be considered too;
- tagging n-grams with ontology classes retrieved at the EL step may be an impactful additional feature;
- the gazetteer is currently being matched at the token level, but it may be more useful if run over the whole input (sentence);
- in order to reduce the noise in the training set, we foresee to leverage a sentence splitter and extract 1-sentence examples only;
- further evaluation experiments will also count EL surface forms instead of links;
- the inclusion of the frame confidence would further refine the final confidence score.

4.12 Conclusion

In a Web where the profusion of unstructured data limits its automatic interpretation, the necessity of *Intelligent Web-reading Agents* turns more and more evident. These agents should preferably be conceived to browse an extensive and variegated amount of Web sources corpora, harvest structured assertions out of them, and finally cater for target KBs, which can attenuate the problem of information overload. As a support to such vision, we have outlined two real-world scenarios involving general-purpose KBs:

(a) Wikidata would benefit from a system that reads reliable third-party resources, extracts statements complying to the KB data model, and leverages them to validate existing data with reference URLs, or to recommend new items for inclusion. This would both improve the overall data quality and, most importantly, underpin the costly manual data insertion and curation flow;

(b) DBpedia would naturally evolve towards the extraction of unstructured Wikipedia content. Since Wikidata is designed to be the hub for serving structured data across Wikimedia projects, it will let DBpedia focus on content besides infoboxes, categories and links.

In this chapter, we presented a system that puts into practice our fourfold research contribution: first, we perform (1) *N-ary relation extraction* thanks to the implementation of Frame Semantics, in contrast to traditional binary approaches; second, we (2) *simultaneously enrich both the T-Box and the A-Box* parts of our target KB, through the discovery of candidate relations and the extraction of facts respectively. We achieve this with a (3) *shallow layer of NLP* technology, namely grammatical analysis, instead of more sophisticated ones, such as syntactic parsing. Finally, we ensure a (4) *fully supervised* learning paradigm via an affordable *crowdsourcing* methodology.

Our work concurrently bears the advantages and leaves out the weaknesses of RE and OIE: although we assess it in a closed-domain fashion via a use case (Section 4.2), the corpus analysis module (Section 4.4) allows to discover an exhaustive set of relations in an open-domain way. In addition, we overcome the supervision cost bottleneck trough crowdsourcing. Therefore, we believe our approach can represent a trade-off between open-domain high noise and closed-domain high cost.

The Relation Extractor is a full-fledged Information Extraction NLP pipeline that analyses a natural language textual corpus and generates structured machine-readable assertions. Such assertions are disambiguated by linking text fragments to entity URIs of the target KB, namely DBpedia, and are assigned a confidence score. For instance, given the sentence `Buffon plays for Serie A club Juventus since 2001`, our system produces the following dataset:

```
@prefix dbpedia: <http://it.dbpedia.org/resource/> .
@prefix dbpo: <http://dbpedia.org/ontology/> .
@prefix fact: <http://fact.extraction.org/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

dbpedia:Gianluigi_Buffon
   dbpo:careerStation dbpedia:CareerStation_01 .

dbpedia:CareerStation_01
   dbpo:team dbpedia:Juventus_Football_Club ;
   fact:competition dbpedia:Serie_A ;
   dbpo:startYear "2001"^^xsd:gYear ;
   fact:confidence "0.906549"^^xsd:float .
```

We estimate the validity of our approach by means of a use case in a specific domain and language, i.e., soccer and Italian. Out of roughly $52,000$ Italian Wikipedia articles describing soccer players, we output more than $213,000$ triples with an estimated average 81.27% F_1. Since our focus is the improvement of existing resources rather than the development of a standalone one, we integrated these results into the Italian DBpedia

CHAPTER[33] and made them accessible through its SPARQL endpoint. Moreover, the codebase is publicly available as part of the DBPEDIA ASSOCIATION repository.[34]

We have started to expand our approach under the Wikidata umbrella, where we feed the *primary sources* tool. The community is currently concerned by the trustworthiness of Wikidata assertions: in order to authenticate them, they should be validated against references to external Web sources. Under this perspective, we are leading the STREPHIT Wikimedia IEG project[35] builds upon the RELATION EXTRACTOR and aims at serving as a reference suggestion mechanism for statement validation. To achieve this, we have successfully managed to switch the input corpus from Wikipedia to third-party corpora and translated our output to fit the Wikidata data model. The soccer use case has already been partially implemented: we have ran the baseline classifier and generated a small demonstrative dataset, named FBK-STREPHIT-SOCCER, which has been uploaded to the primary sources tool back-end. We invite the reader to play with it, by following the instructions in the project page.[36] At the time of writing this article, we are scaling up to (a) a larger input in (b) the English language, with (c) a bigger set of relations, and (d) a different domain. The *Web Sources* corpus contains more than 500,000 English documents gathered from 53 sources; the corpus analysis yielded 50 relations, which are connected to an already available frame repository, i.e., FrameNet.

For future work, we foresee to progress towards multilingual open information extraction, thus paving the way to (a) its full deployment into the DBpedia Extraction Framework, and to (b) a thorough referencing system

[33] http://it.dbpedia.org/2015/09/meno-chiacchiere-piu-fatti-una-marea-di-nuovi-dati-estratti-dal-testo-di-wikipedia/?lang=en

[34] https://github.com/dbpedia/fact-extractor

[35] https://meta.wikimedia.org/wiki/Grants:IEG/StrepHit:_Wikidata_Statements_Validation_via_References

[36] https://www.wikidata.org/wiki/Wikidata:Primary_sources_tool#How_to_use

for Wikidata.

Acknowledgments. The Relation Extractor has been developed within the DBpedia Association and was partially funded by Google under the Summer of Code 2015 program. The StrepHit project is undergoing active development and is fully funded by the Wikimedia Foundation via the Individual Engagement Grants program.

Algorithm 1 Rule-based baseline classifier

Input: S; F; L

Output: C

$C \leftarrow \emptyset$
for all $s \in S$ **do**
  $E \leftarrow entityLinking(s)$
  $T \leftarrow tokenize(s)$
  for all $t \in T$ **do**
    if $t \in L$ **then** #Check whether a sentence token matches a LU token
      for all $f \in F$ **do**
        $core \leftarrow$ **false**
        $O \leftarrow getLinkedEntityClasses(E)$
        for all $o \in O$ **do**
          $fe \leftarrow lookup(f)$ #Get the FE that maps to the current linked entity class
          $core \leftarrow checkIsCore(fe)$
        end for
        if $core$ **then** #Relaxed classification
          $c \leftarrow [s, f, fe]$
          $C \leftarrow C \cup \{c\}$
        else
          continue #Skip to the next frame
        end if
      end for
    end if
  end for
end for
return C

Chapter 5

Classes: Unsupervised Taxonomy Learning

5.1 Introduction

The Wikipedia category system is a fine-grained topical classification of Wikipedia articles, thus being natively suitable for encoding Wikipedia knowledge. Besides its ontology, DBpedia uses the category hierarchy as a supplementary classification system, while several taxonomization efforts such as [102, 103, 34, 49, 85, 86, 63], aim at mapping categories into types. However, their granularity is often very high, resulting in an arguably overly large set of items. From a practical perspective, it is vital to cluster resources into classes with intuitive labels, in order to simplify the end user's cognitive effort needed when querying the knowledge base. Hence, identifying a taxonomy based on a prominent subset of Wikipedia categories is a critical step to both extend and homogenize the DBpedia ontology (DBPO).

Despite the number of similar initiatives, we argue that there is a need for a dataset with broad coverage and satisfactory intuitiveness. In this chapter, we present *DBTax*, a completely data-driven methodology to automatically construct a comprehensive classification of DBpedia resources.

Four features set DBTax apart from related approaches and constitute the main contributions of this chapter:

1. Exhaustive type coverage over the whole knowledge base;
2. Focus on the actual usability of the schema from an end user's perspective;
3. Possibility of replication across different Wikipedia language chapters;
4. Fully unsupervised implementation, not requiring manual efforts for building annotated corpora.

The remainder of this chapter is structured as follows. We first outline in section 5.2 a high-level overview of the approach, with a definition of the key intuition. section 5.3 contains our core contribution and illustrates in detail its major implementation phases. We corroborate our methodology with a report of its outcomes (section 5.4), coverage comparisons with related resources, as well as an evaluation of both the taxonomy structure and the type assignment correctness (section 8.7). In section 5.6, we describe the policies to ensure access and sustainability of the output datasets, before drawing our conclusions in section 5.7.

5.2 Prominent Nodes

We propose to automatically derive a taxonomy for the classification of DBpedia resources from a prominent subset of the Wikipedia category system, which provides a more reliable and almost complete knowledge backbone compared to infoboxes. We report below a high-level overview of our prominent node identification core algorithm, with the help of an example. A detailed description is provided in subsection 5.3.2. The category with label Media in Traverse City, Michigan has 2 subcategories, namely

(a) Radio stations in Traverse City, Michigan (mentioned in 8 pages), and (b) Television stations in Traverse City, Michigan (mentioned in 4 pages). Both subcategories are leaf nodes. Thus, we make the parent category a *prominent node* and organize the 12 pages into a single cluster. Since this algorithm solely considers the category system structure, we incorporate linguistic processing and a usage-based technique. The former aims at simplifying the cluster label, which is renamed to Media in our example. The latter weights the cluster depending on how often it is employed across all the Wikipedia language chapters.

5.3 Generating DBTax

We envision the construction and the population of DBTax in four major stages:

1. Leaf node extraction;
2. Prominent node discovery;
3. Class taxonomy generation (T-Box);
4. Pages type assignment (A-Box).

First, we describe in subsection 5.3.1 a method to identify initial leaf node candidates. In subsection 5.3.2, we provide an overview of the prominent node discovery procedure step by step. The algorithms used to generate the class hierarchy are illustrated in subsection 5.3.3. Finally, we assign types to Wikipedia pages (subsection 5.3.4).

5.3.1 Stage 1: Leaf Nodes Extraction

The Wikipedia category system is organized in a cyclic graph data structure, which is of little use from a taxonomical perspective, due to its noisy nature.

In fact, a class hierarchy best fits into a directed acyclic graph (DAG) data structure, and we adopt a bottom-up approach to build it, starting from the leaves up to the root. Hence, the first stage takes as input the Wikipedia public database dumps[1] and outputs a set of *leaf nodes*, i.e., categories with no subcategories, which we store in a database table (NODE). Specifically, we use the Wikipedia tables encoding the links between the categories themselves, as well as between the categories and the pages. The procedure is implemented as follows: (a) we retrieve the full set of article pages, (b) we extract those categories that are linked to actual articles only, by looking up the outgoing links for each page, and out of them (c) we determine the set of categories with no subcategories.

5.3.2 Stage 2: Prominent Node Discovery

The following techniques are combined to identify the set of prominent category nodes:

1. *Algorithmic*, programmatically traversing the Wikipedia category system;

2. *Linguistic*, identifying categories yielding is-a relations via Natural Language Processing;

3. *Multilingual*, leveraging interlanguage links.

The algorithmic technique is launched first and its output serves the other ones in a parallel fashion. We implement their outcomes in the form of attributes in the NODE database table, where a category represents a record.

[1] https://dumps.wikimedia.org

Algorithm 2 Prominent Node Discovery

Input: L **Output:** $PN \neq \varnothing$

1: $PN \leftarrow \varnothing$
2: **for all** $l \in L$ **do**
3: $isProminent \leftarrow$ **true**; $P \leftarrow getTransitiveParents(l)$
4: **for all** $p \in P$ **do**
5: $C \leftarrow getChildren(p)$; $areAllLeaves \leftarrow$ **true**
6: **for all** $c \in C$ **do**
7: **if** $c \notin L$ **then** $areAllLeaves \leftarrow$ **false**; **break**
8: **end for**
9: **if** $areAllLeaves$ **then**
10: $PN \leftarrow PN \cup \{p\}$; $isProminent \leftarrow$ **false**
11: **end for**
12: **if** $isProminent$ **then** $PN \leftarrow PN \cup \{l\}$
13: **end for**
14: **return** PN

Traversing the Leaf Graph

We now illustrate the procedure to programmatically process the Wikipedia category graph, starting from the set of leaf nodes produced in subsection 5.3.1 and yielding a set of prominent node candidates. Its pseudocode is provided in Algorithm 2. The approach can be resumed as follows. Given as input a set of leaf nodes L, for each leaf l, we transitively traverse back to its set of parents P. For each such parent p, we check whether its set of children C is exclusively composed of leaves. If so, we consider p a prominent node and add it to the output set PN. Otherwise, we make l a prominent node. We use a boolean attribute to mark PN elements in the NODE table.

NLP for is-a Relations

We adopt the approach applied in YAGO [63, 118] to identify prominent node candidates holding *is-a* relations. It relies on a straightforward yet

powerful observation: since any Wikipedia category linguistically corresponds to a noun phrase, if its head appears in plural form, then that category is likely to be a conceptual one, and may serve as a class (cf. the paragraph on YAGO in Section 2.5.1). Specifically, we perform shallow syntactic parsing by means of the Noun Group Parser [117]. Categories are represented via link grammars [115], which are simple implementations of phrase structure grammars, the most complex being HPSG [101, 100].

For instance, Figure 5.1 explains how to parse the noun phrase (NP) PAST PRESIDENTS OF ITALY, which yields 3 chunks, namely a pre-modifier (PRE) PAST, a *head* PRESIDENTS and a post-modifier (POST) OF ITALY. We populate a new attribute of the NODE table with the head chunk. Afterwards, we exploit the Pling-Stemmer[2] to automatically mark prominent nodes having a plural head with a boolean attribute. The replicability of such method across multilingual Wikipedia deployments can be achieved via the following two strategies, each bearing its price: (a) exploitation of category interlanguage links (published by Wikipedia), at the cost of excluding categories with no English counterpart, and (b) language-specific implementations of the noun phrase parser and the stemmer, both at an intrinsic development expense and depending on the availability of language resources.

Interlanguage Links as a Weight

We leverage the LANGLINKS table of the Wikipedia database dumps to retrieve the number of interlanguage links for each prominent node candidate. This enables the implementation of a usage-driven weighting system, since we are able to induce a score assessing the usage of a given category among all the Wikipedia language editions. We populate a further attribute of the

[2] http://resources.mpi-inf.mpg.de/yago-naga/javatools/doc/javatools/parsers/PlingStemmer.html

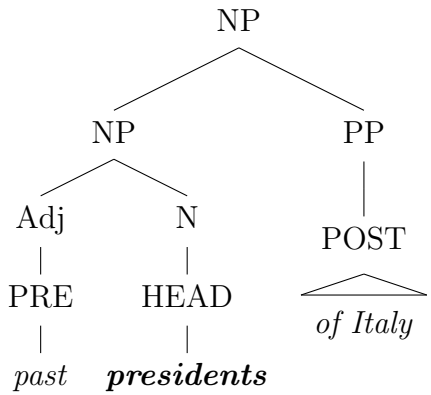

Figure 5.1: Example of a Wikipedia category phrase structure parsing tree

NODE table with the interlanguage links weight, and use it as a threshold to filter out underutilized items.

5.3.3 Stage 3: Class Taxonomy Generation

We reconstruct the full hierarchy of parent-child relations by recursively obtaining the set of parents for each leaf category, following a bottom-up direction.

Cycle Removal

The Wikipedia category graph contains cycles and so did the output of our first reconstruction attempt. In order to remove them and ensure a strict hierarchy, we apply Algorithm 3 in our processing pipeline. In brief, the algorithm traverses the graph in a breadth-first top-down fashion, starting from the root node (i.e., CONTENTS) and returns a tree T. For each node we encounter, we add it to T only if it has not been introduced yet. The set E keeps the already introduced nodes, while sets P and N keep the nodes for a specific tree level. The breadth-first approach for cycle removal favors shorter hierarchy paths: if a category exists in multiple levels of the graph, the node with the lowest depth will be added with a low distance to

the root. However, we believe this choice both satisfies the goals of DBTax and complies with the philosophy of DBPO, namely to provide a high-level and general-purpose classification.

Pruning instances

The taxonomy we have obtained from the methods applied so far does not make the distinction between classes and instances. Thus, we need to leverage further post-processing to prune instances and to produce a consumable resource. We opt for the name analysis approach proposed in [130], which assumes that instances are real-world entities. We leverage the DBpedia 3.9 release to filter out non-classes. Specifically, we combine the datasets containing labels, redirects and instances, and generate a list of labels for all DBpedia instances. By joining this list with the taxonomy, we managed to exclude 1,562 entries. Even though the pruning step cleaned DBTax from instances, it additionally removed many nodes from the hierarchy. This unavoidable side-effect partially decreased the quality of the T-Box. The reason is that nodes with pruned parents got attached directly to the root, thus resulting in broad paths (cf. section 5.4).

5.3.4 Stage 4: Pages Type Assignment

We populate the taxonomy built in stage 3 by taking as input the heads of the prominent nodes returned in stage 2 and by leveraging the links between categories and Wikipedia article pages. In this way, we are able to assert an *instance-of* relation between a given page and the head of a category linked to that page. Once the type is assigned, its super and subtypes can be automatically inferred on account of the T-Box. We informally report below the foreseen procedure, which is applied to each prominent node head h.

Algorithm 3 Cycle Removal

Input: G **Output:** $T \neq \varnothing$

```
1: T ← ∅; P ← getRootNode(G); E ← P
2: while P ≠ ∅ do
3:    N ← ∅
4:    for all p ∈ P do
5:       C ← getChildren(p)
6:       for all c ∈ C do
7:          if c ∉ E then
8:             E ← E ∪ {c}; N ← N ∪ {c}; T ← T ∪ {p, c}
9:       end for
10:   end for
11:   P ← N
12: end while
13: return T
```

1. Extract the set S of those categories having head $= h$;
2. Extract the pages linked to each category in S;
3. For each page p:
   (a) If it is an article page, then produce an assertion in the form of a triple $< p, \textit{instance-of}, h >$
   (b) If it is a category, recursively repeat from point 2 until the condition in point 3(a) is satisfied.

5.4 Results

In order to enable the comparison across related resources, we process the same April 2013 English Wikipedia dumps as the DBpedia 3.9 release.[3] The outcomes of DBTax are three-fold, namely:

[3] http://wiki.dbpedia.org/services-resources/datasets/data-set-39/dump-dates-39

- The taxonomy (T-Box) automatically generated according to stage 3 (subsection 5.3.3) is composed of 1,902 classes;

- 10,729,507 *instance-of* assertions (A-Box) are produced as output of stage 4 (subsection 5.3.4). They are serialized into triples, according to the RDF data model.[4] We use the Turtle[5] syntax, which supports UTF-8-encoded International Resource Identifiers (IRIs), thus fitting well for multilingual Wikipedia pages with no need for escaping special characters. An example is reported as follows.

  dbpedia:Combat_Rock a dbtax:Album .

- A total of 4,260,530 unique resources are assigned a type, 2,325,506 of which do not have one in the DBpedia 3.9 release.

5.5 Evaluation

We use the following versions of the resources we compare to: (a) DBPO version 3.9;[6] (b) MENTA's underlying Wikipedia dumps date back to 2010; (c) SDType as per DBpedia version 3.9;[7] (d) YAGO types dataset as per DBpedia version 3.9;[8] (e) WiBi consumes the October 2012 English Wikipedia dump;[9] (f) Wikipedia categories from the same April 2013 English Wikipedia dumps; (g) Wikidata RDF exports from April 2014.[10] We decided to insert both MENTA and WiBi into our comparative evaluation anyway, since the former leverages knowledge from 271 languages, and the latter stands as the most recently published (2014) related approach. However, we

[4] http://www.w3.org/TR/rdf11-concepts/
[5] http://www.w3.org/TR/turtle/
[6] http://downloads.dbpedia.org/3.9/dbpedia_3.9.owl.bz2
[7] http://downloads.dbpedia.org/3.9/en/instance_types_heuristic_en.ttl.bz2
[8] http://downloads.dbpedia.org/3.9/links/yago_types.ttl.bz2
[9] http://wibitaxonomy.org/wibi-ver1.0.tar.gz
[10] http://tools.wmflabs.org/wikidata-exports/rdf/exports/20140420/

recognize their performance might be relatively different on the April 2013 dump. Furthermore, the closest Wikidata dump we could access is one year newer. Hence, we expect a performance variation there as well. Finally, we could not retrieve the T-Box from MENTA and SDType, thus limiting their evaluation to the A-Box only. We could not build our experiments with Tìpalo [51], since the only available dataset[11] contains 547 unique entities, and has no overlap with our evaluation sets (cf. Section 5.5.2 and 5.5.3).

5.5.1 Coverage

Exhaustive type coverage over the whole knowledge base is a crucial objective in our contribution. We compute coverage as the number of resources for which at least one type is assigned, divided by the amount of actual Wikipedia article pages in the dump we process, excluding *redirect* pages. We report the values in Table 5.1. DBTax clearly outperforms all the compared resources. Since our approach depends on the Wikipedia categories, one may object that articles with no assigned categories cannot be covered. However, at the time of writing this paper (August 2015), merely 2,263 English Wikipedia articles are uncategorized[12] (exclusively considering *content* categories, not *administrative* ones).[13] This corresponds to circa 0.045% of the total 4,934,195 articles.[14] Hence, the results we obtained for DBTax are in line with the statistics reported by the English Wikipedia. Moreover, DBTax identified 20.6% of DBPO manually curated classes, ranging from top-level (e.g., Work), to deeply nested (e.g., Biomolecule) ones. Such finding enables a natural mapping to DBPO.

[11] http://ontologydesignpatterns.org/ont/wikipedia/instance.rdf
[12] http://en.wikipedia.org/wiki/Category:All_uncategorized_pages
[13] http://en.wikipedia.org/wiki/Wikipedia:Categorization#Non-article_and_maintenance_categories
[14] http://meta.wikimedia.org/wiki/List_of_Wikipedias

5.5.2 T-Box Evaluation

We compare our results against DBPO, YAGO, WiBi, and Wikidata class hierarchies, as well as the Wikipedia category system itself, treating the Wikipedia categories as classes for the purpose of this evaluation only. We focus on (1) distinguishing classes from instances, and (2) hierarchy paths.

Task Anatomy

We pick a random sample of 50 classes from each resource and ask the evaluators the following questions: (a) *"Is this a class or an instance?"* (Class), and (b) *"Can this class be broken down into more than one class?"* (Breakable). For the hierarchy path evaluation, we pick a random sample of 50 *leaf* classes from each resource and generate the hierarchy path up to the root node (i.e., THING). We ask the evaluators the following questions: (a) *"Is this a valid class hierarchy path?"* (Valid), (b) *"Is this hierarchy too specific?"* (Specific), and (c) *"Is this hierarchy too broad?"* (Broad). The *Valid* question is meant to catch wrong hierarchies (e.g., THING ➤ CITY ➤ PLACE). The *Specific* and *Broad* questions aim at capturing such taxonomy design issues, although we recognize that they can be subjective and may depend on the use case. In fact, we expect a low agreement score, as we are assessing general-purpose taxonomies, with a high probability of

Table 5.1: Type coverage of Wikipedia articles

Resource	Coverage
DBPO	.513
DBTax	**.994**
MENTA	.537
SDType	.147
YAGO	.673
WiBi	.794

cross-domain knowledge in our evaluation set. The *Breakable* and *Specific* questions involve leaf nodes only, while *Valid* is formulated with a path from a leaf node to the root. In total, 10 evaluators participated and each question was evaluated twice. The namespaces were hidden to avoid bias and the questions were globally randomized.

Discussion

Table 5.2 shows the overall results. Out of the four taxonomies, DBPO averagely performs slightly better. However, we expected such behavior, since it is a relatively small and manually curated ontology, compared to YAGO and DBTax. YAGO yields similar results to DBTax with respect to the *Valid* question. DBTax provides better non-breakable classes, as it solely consists of prominent nodes and does not create too specific hierarchies (cf. *!S*), as opposed to YAGO. Finally, DBTax stands last when it comes to broad hierarchies (cf. *!B*). This is due both to the cycle removal algorithm and especially to the instance pruning step (cf. subsection 5.3.3), where several nodes were removed and leaf nodes got attached to the root. The main cause is the massive presence of instances in Wikipedia categories. The way we propose to overcome this is to outsource DBTax to the DBpedia ontology community and allow the community to perform the alignment. Although the *!Specific* and *!Broad* questions seem complementary, our intention is to additionally identify average hierarchy paths, suitable for a general-purpose taxonomy.

5.5.3 A-Box Evaluation

Assessing the actual usability of our knowledge base has the highest priority in our work. Moreover, estimating the quality of the assigned types must cope with subjectivity issues, as emphasized in [118]. Therefore, we decided to adopt an online evaluation approach with common users. Under this

Table 5.2: T-Box evaluation results. *C* is the ratio of classes in the taxonomy and *!Bre* the ratio of classes that cannot be broken into other classes. *V* is the ratio of valid hierarchy paths, *!S* the ratio of paths that are not too specific, and *!Bro* the ratio of paths that are not too broad

	C	**!Bre**	**V**	**!S**	**!Bro**
DBPO	.66	.67	.89	.97	.84
DBTax	.65	.76	.77	.98	.40
YAGO	.90	.38	.81	.55	.93
WiBi	.75	.38	.73	.41	.85
Wikidata	.19	.48	.85	.66	.88
Wikipedia	.81	.29	.66	.77	.78
Fleiss' κ	.32	.23	.23	.06	.30

perspective, the major issue consists of gathering a sufficiently heterogeneous amount of judgments. Micro-payment services represent a suitable solution, since they allow us to outsource the evaluation task to a worldwide massive community of paid workers. We leverage the CrowdFlower platform,[15] which serves as a bridge to a plethora of crowdsourcing channels. In this way, we are able to simultaneously determine (a) the cognitive correctness of the assertions, and (b) the intuitiveness of the underlying semantics.

Task Anatomy

We randomly isolate 500 entities from those that do not have a type counterpart in DBpedia. Hence, we consider our evaluation set to be representative of the problem we are trying to tackle, namely to provide extensive classification coverage for DBpedia. While building our task, we aim at maximizing ease and atomicity. Workers are shown (1) a link to a Wikipedia page (i.e., the entity itself), labeled with the word *this* in the question *"What is this?"*, and (2) a type (i.e., the object of the instance-of relation, such as BAND), rendered in the form *"Is it a {type}?"*. Then, they

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

are asked to (1) visit the page, and (2) judge whether the type is correct, by answering a *Yes/No* question.

For each entity, we elicit 5 judgments, thus gathering a total of 2,500. We prevent each worker from answering a question more than once by setting 500 maximum judgments per contributor and per IP. Finally, we ensure that all countries are allowed to work on our task and set the payment per page to \$.03, where a page contains 5 entities. A cheating check mechanism is implemented via test questions, for which we supply the correct answer in advance. If a worker misses too many test answers within a given threshold (80% in our case), he or she will be banned and his or her *untrusted* judgments will be automatically discarded.

Table 5.3: Comparative A-Box evaluation on 500 randomly selected entities with no type coverage in DBpedia. ♠ indicates statistically significant difference with $p < .0005$ using χ^2 test, between DBTax and the marked resources

Resource	**P**	**R**	**F_1**	**Agr**	**Untrusted**
DBTax	.744	1	.853	.857	**518**
MENTA	.793	.589♠	.675	.826	1,093
SDType	.924	.098♠	.178	.899	1,723
YAGO	.461♠	.727	.565	.868	1,358
WiBi	.858	.597	.704♠	.924	2,075
Wikidata	.808	.982	.886	.913	1,847

Discussion

CrowdFlower provides a full report with detailed information for every single judgment made on the platform. For each question, an agreement score computed via majority vote weighted by worker trust is also included, and we calculate the average among the whole evaluation set. Table 5.3 displays the results obtained by processing the report. We compute precision as the ratio between positive answers and the total amount of answers, and

recall as the ratio between positive answers and the sum of positive answers with the untyped entities (multiplied by 5 missing judgments). First, we notice that all resources are affected by recall issues, since they have a lack of type information, while our approach is always able to assign a type. This corroborates our findings on type coverage as per Table 5.1, where our system almost achieves 100%, in strong contrast to the other resources. To our surprise, DBTax also remarkably outperforms YAGO in terms of precision (validated by a statistical significance test), while the other resources generally behave better, although at a high recall cost. In a nutshell, DBTax scores satisfactorily high precision while reaching full recall. Via this trade-off, it achieves the best F_1 value, compared to automatically generated resources. Wikidata obtains the absolutely highest F_1, but we believe this might be due to the heavy manual curation efforts of millions of human contributors.[16]

Given similar agreement values (cf. the *Agr* column), the number of untrusted judgments may be viewed as a further indicator of the overall question ambiguity. In fact, we tried to maximize objectivity and simplicity when choosing test questions. However, it is known that the choice of taxonomical terms is always controversial, even for handcrafted taxonomies. Since the entities are identical in all the experiments, we can infer that the number of workers who missed the tests is directly influenced by the type ambiguity, which is the only variable parameter. In the light of the tangible discrepancy between the untrusted judgments values, we claim that DBTax is much more intuitive from a cognitive ergonomics perspective, even for common worldwide end users.

[16] https://www.wikidata.org/wiki/Special:Statistics

5.6 Access and Sustainability

DBTax datasets will be included in the next and all subsequent official DBpedia releases. Within the release, it will serve as a complementary set of A- and T-Box statements to structure DBpedia resources. Thanks to the natural mapping to DBPO, an A-Box subset containing DBPO type assertions only is made available as well.[17] The first DBpedia release (v. 2015A) that will include this dataset is due on mid 2015. Since DBpedia is a pioneer in adopting and creating best practices for RDF publishing, being incorporated into its workflow guarantees regular updates. Long-term availability will be ensured through the DBpedia Association and the Leipzig Computing Data Center.

Until DBTax is not served by the regular DBpedia releases, the dataset is hosted at the *Italian DBpedia* chapter.[18] Moreover, it is registered on DataHub[19] and VOID metadata[20] is provided. Since DBTax is part of the official DBpedia releases, it benefits from the same users and developers communities, as well as support infrastructure.

5.7 Conclusion

DBTax is the outcome of a completely data-driven approach to convert the chaotic Wikipedia category system into an extensive general-purpose taxonomy. As a result of our four-step processing pipeline, we generated a hierarchy of 1,902 classes and automatically assigned types to roughly 4.2 million DBpedia resources. Thus, we provide a significant coverage leap, as opposed to DBpedia (with only 2.2 million typed resources) and to related automatic approaches. Moreover, online evaluations in a crowdsourcing

[17] http://it.dbpedia.org/downloads/dbtax/A-Box-dbpo.nt.bz2
[18] http://it.dbpedia.org/downloads/dbtax/
[19] http://datahub.io/dataset/dbpedia-dbtax
[20] http://it.dbpedia.org/downloads/dbtax/void.ttl

environment demonstrate that DBTax is not only comparable to the manually curated DBpedia ontology (DBPO) in terms of taxonomical structure, but is also outstandingly intuitive for common end users, while achieving the best precision and recall trade-off. DBTax is currently deployed in the Italian DBpedia chapter SPARQL endpoint[21] and will be included in all future DBpedia releases. We envision DBTax to serve as a balance between DBPO and YAGO, as we argue that DBPO is very limiting and YAGO far too large for real-world use cases.

[21] http://it.dbpedia.org/2015/02/dbpedia-italiana-release-3-4-wikidata-e-dbtax/?lang=en

Chapter 6

Application: Knowledge Base-driven Recommender Systems

6.1 Introduction

Recommender systems try to tackle the problem of information overload by offering personalized suggestions. They play today a crucial role in several applications, ranging from e-commerce to news portals, all the way to enterprise information management systems. While their performance is confirmed and their use is widespread, we aim at investigating the role of large-scale richly structured knowledge bases in the recommendation process. News recommendation is a real-world application of such systems and is growing as fast as the online news reading practice: it is estimated that, in May 2010, 57% of U.S. Internet users consumed online news by visiting news portals [76]. Recently, online news consumers seem to have changed the way they access news portals: "just a few years ago, most people arrived at our site by typing in the website address. (...) Today the picture is very different. Fewer than 50% of the 8 million+ visitors to the News website every day see our front page and the rest arrive directly at a story", a product manager of the BBC News website affirms,[1] indicating

[1] http://www.bbc.co.uk/blogs/bbcinternet/2012/03/bbc_news_facebook_app.html

the need for news information filtering tools.

The online reading practice leads to the so-called *post-click* news recommendation problem: when a user has clicked on a news link and is reading an article, he or she is likely to be interested in other related articles. This is still a typical editor's task, namely an expert who manually looks for relevant content and builds a recommendation set of links, which will be displayed below or next to the current article. The primary aim is to keep users navigating on the visited portal. News recommender systems attempt to automate such task. Current strategies can be clustered into 3 main categories [65], namely (a) collaborative filtering, (b) content-based recommendation, and (c) knowledge-based recommendation. (a) focuses on the similarities between users of a service, thus relying on user profiles data. (b) leverages term-driven information retrieval techniques to compute similarities between items. (c) mines external data to enrich item descriptions.

In this chapter, we propose a novel news recommendation strategy, which leverages both NLP techniques and semantically structured data. We show that entity linking tools can be coupled to existing knowledge bases in order to compute unexpected suggestions. Such knowledge bases are used to discover meaningful relations between entities. As a preliminary work to assess the validity of our approach, we focus on a celebrity gossip use case and consume data from the TMZ news portal and Freebase.[2] For instance, given a TMZ article on `Michael Jackson`, our strategy is able to detect from Freebase that `Michael Jackson` (a) is a dead celebrity who had drug problems and (b) dated with `Brooke Shields`, thus suggesting other TMZ articles on `Amy Winehouse`, `Kurt Cobain` (other dead celebrities who had drug problems) and `Brooke Shields`. We investigate if user attention can be attracted via specific explanations, which clarify why a

[2] http://www.tmz.com, http://www.freebase.com/

given recommendation set is proposed. Such explanations are built on top of the entity relations. Finally, we conducted an online evaluation with real users. We outsourced a set of experiments to the community of paid workers from Amazon's Mechanical Turk (AMT) crowdsourcing service.[3] The collected results confirm the effectiveness of our approach.

Our primary aim is to attract the attention of a generic user, since post-click news recommendation generally relies on a single click user profile data. Therefore, we are set apart from most traditional recommender systems with respect to three main features:

1. *User agnosticity*: user interests are deduced from user profile data and contribute to the quality of recommendations. Collecting explicit feedback is a costly task, as it requires motivated users. Our approach gives low priority to user profiles.

2. *Unexpectedness*: similarity, novelty and coherence are key components for satisfactory news recommendations [76]. Content-based strategies tend to propose too similar items and create an 'already seen' sensation. We believe entity relations discovery can augment both novelty and coherence, thus leading to unexpected suggestions.

3. *Specific explanation*: in news web portals, generic sentences such as `Related stories` or `See also` are typically shown together with the recommendation set. We expect that more specific sentences can improve the trustworthiness of the system.

6.2 Approach

Our strategy merges content-based and knowledge-based approaches and is defined as a *hybrid entity-oriented* recommendation strategy enhanced by

[3] https://www.mturk.com/mturk/welcome

human-readable explanations. Given a source article from a news portal, we recommend other articles from the portal archive, namely the corpus, by leveraging both entity linking techniques and knowledge extraction from semantically structured knowledge bases. Specifically, we gathered a celebrity gossip corpus from TMZ and chose Freebase as the knowledge base.

We consider both the corpus and the knowledge base as a unique object, namely a *dataspace*, which results from heterogeneous data sources integration. Each data source is converted into an RDF graph and becomes an element of the dataspace. Such dataspace can then be queried in order to retrieve sets of recommendations. A *semantic recommender* exploits SPARQL graph navigation capabilities to output recommendation sets. Each recommender is built on top of a concept, e.g., *substance abuse*.

The entity linking step in the corpus processing phase enables the detection of both real-world entities and encyclopedic concepts. We compute concept statistics on the whole corpus and assume that the most frequent ones are likely to generate interesting recommendations. A mapping between corpus concepts and meaningful relations of the knowledge base allows the creation of recommenders. Table 6.1 shows the TMZ-to-Freebase n-ary concept mapping we manually built. Each Freebase value represents the starting point for the construction of a recommender, while the string after the last dot becomes the name of the recommender, e.g., *parents*.

Given an entity of the source article, a name of a recommender and an entity contained in the recommendation sets, we are able to construct a specific explanation. Ultimately, a ranking of all the recommendation sets produces the final top-N suggestions output.

Table 6.1: TMZ-to-Freebase mapping

TMZ	Freebase
Family	people.person.{parents, sibling_s, children, spouse_s}
Intimate_relationship	celebrities.celebrity.sexual_relationships
Dating	base.popstra.celebrity.dated
Ex_(relationship)	base.popstra.celebrity.breakup
Net_worth	celebrities.celebrity.net_worth
Substance_abuse	celebrities.celebrity.substance_abuse_problems
Conviction	base.crime.convicted_criminal
Court	law.court.legal_cases
Arrest	base.popstra.celebrity.{arrest, prison_time}
Legal_case	law.legal_case.subject
Criminal_charge	celebrities.celebrity.legal_entanglements
Judge	law.judge
Death	people.deceased_person
Television_program	tv.tv_program

6.3 System Architecture

Figure 6.1 describes the general system workflow. The major phases are (a) corpus processing, (b) knowledge base processing, (c) dataspace querying and (d) recommendation ranking.

TMZ Processing Pipeline.

Given as input a set of TMZ articles, we output an RDF graph and load it into the dataspace. Corpus documents are harvested via a subscription to the TMZ RSS feed. The RSS feed returns semi-structured XML documents. A cleansing script extracts raw text from each XML document. The entity linking step exploits *The Wiki Machine*,[4] a state-of-the-art [82] machine learning system designed for linking text to Wikipedia, based on a word sense disambiguation algorithm [54]. For each raw text document, real-world entities such as persons, locations and organizations are recognized, as well as encyclopedic concepts. This enables (a) the assignment of a

[4] http://thewikimachine.fbk.eu

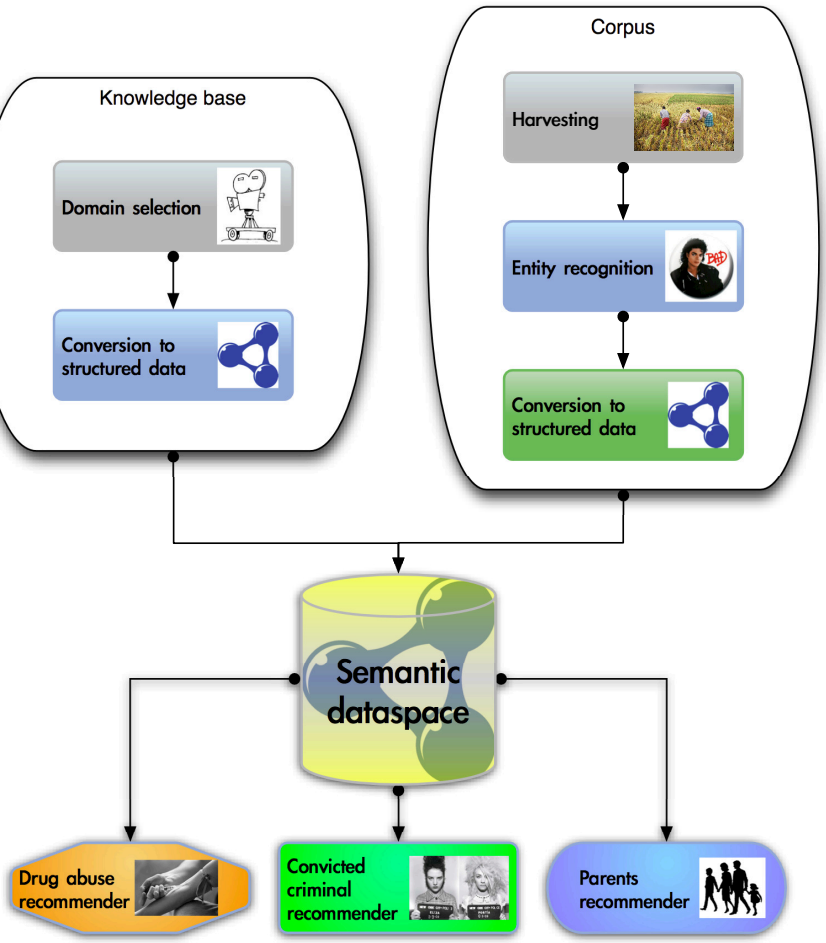

Figure 6.1: High level system workflow

unique identifier, namely a DBpedia URI to each annotation and (b) the choice of top corpus concepts for recommenders building purposes. The Wiki Machine takes a plain text as input and produces an RDFa document.[5] The extracted terms are assigned an `rdf:type`, namely `NAM` for real-world entities or `NOM` for encyclopedic concepts. The `hasLink` property connects the terms to the article URL they belong, thus enabling the computation of the recommendation set. Other metadata, such as the link to the corresponding Wikipedia page and the annotation confidence score are also expressed. RDFa documents are converted into RDF data via the Any23

[5]The full corpus of TMZ RDFa documents is available at http://bit.ly/QLph9B

library.[6] RDF data is loaded into a Virtuoso[7] triple store instance, which serves the dataspace for querying.

Freebase Processing Pipeline.

Freebase provides exhaustive granularity for several domains, especially for celebrities. Given that such knowledge base is large, we avoid loading its complete version, because of severe performance issues we encountered. Consequently, meaningful slices corresponding to the corpus domains, e.g., celebrities, people, are selected. A domain-dependent subset is then produced via a filter written in Java. The dataset is converted into RDF data with logic implemented in Java. Finally, RDF data is loaded into a Virtuoso triple store instance.

6.3.1 Querying the Dataspace

A recommender performs a join between an entity belonging to the TMZ graph and the corresponding entity belonging to the Freebase graph. TMZ entities are identified by a DBpedia URI, which differs from the Freebase one. Therefore, we exploit `sameAs` links between DBpedia and Freebase URIs. Recommenders are divided in two categories, namely (a) *entity-driven* and (b) *property-driven*.[8] For each detected entity of the source article, we run Freebase schema inspection queries[9] and retrieve its types and properties. Thus, we are able to recognize which recommenders can be triggered for a given entity. Building a recommender requires (a) knowledge of relevant Freebase schema parts in order to properly browse its graph and (b) a sufficiently expressive RDFa model for named entities and link retrieval. The `NAM` type and the `hasLink` property provide such expressivity.

[6] http://incubator.apache.org/any23/
[7] http://virtuoso.openlinksw.com/
[8] The full sets are available at http://bit.ly/MWGu06 and http://bit.ly/MWGsW3
[9] Available at http://bit.ly/MVGVtE

Entity-Driven Recommenders.

The queries behind entity-driven recommenders contain an `%entity%` parameter that must be programmatically filled by an entity belonging to the source article. For instance, given an article in which `Jessica Simpson` is detected and triggers the *sexual relationships* recommender, we are able to return all the corpus articles (if any) that mention entities who had sexual relationships with her, e.g., `John Mayer`. To avoid running empty-result recommenders, we built a set of `ASK` queries,[10] which check if recommendation data exists for a given entity. The *sexual relationships* query follows:

```
PREFIX fb: <http://rdf.freebase.com/ns/>
PREFIX twm: <http://thewikimachine.fbk.eu#>
SELECT DISTINCT ?had_relationship_with ?link
WHERE { <http://dbpedia.org/resource/%entity%> owl:sameAs ?fb_entity .
?fb_entity fb:celebrities.celebrity.sexual_relationships ?fb_sexual_rel .
?fb_sexual_rel fb:celebrities.romantic_relationship.celebrity ?fb_celeb .
?fb_celeb fb:type.object.name ?had_relationship_with .
?dbp_celeb owl:sameAs ?fb_celeb ; a twm:NAM ; twm:hasLink ?link ; twm:hasConfidence ?conf .
FILTER (?fb_entity != ?fb_celeb) . FILTER (lang(?had_relationship_with)='en') . }
ORDER BY DESC (?conf)
```

Property-Driven Recommenders.

After the schema inspection step, an entity of the source article can directly trigger one of these recommenders if it contains the corresponding property. Property-driven queries return articles that mention entities who share the same property. Hence, they do not require a parameter to be filled. For instance, given an article in which `Lindsay Lohan` is detected and the property `legal entanglements` is identified during the schema inspection step, we can suggest other articles on people who had legal entanglements, e.g., `Britney Spears`.

[10] Available at http://bit.ly/NDNORH

Building Explanations.

Specific explanations are handcrafted from <*s, r, o*> triples, where *s* is a *subject* entity that was extracted from the source article, *r* is the *relation* expressed by the triggered recommender and *o* is an *object* entity for which the recommendation set is computed. Therefore, we are able to construct different explanations depending on the elements we use. For instance, (a) *s,r,o* yields: `Jessica Simpson` had `sexual relationships` with `John Mayer`. Read more about him. (b) *s,r* yields: Read more about `Jessica Simpson`'s `sexual relationships`. (c) *r,o* yields: Read more about her `sexual relationships` with `John Mayer`.

6.3.2 Ranking the Recommendation Sets

Since recommendations originate from database queries, they are unranked and in some cases too many. To overcome the problem, we implemented an information retrieval ranking algorithm and are able to provide top-N recommendations. The bag-of-words (BOW) cosine similarity function is known to perform effectively for topic-related suggestions [95]. However, it does not take into account language variability. Consequently, we also leverage a latent semantic analysis (LSA) algorithm.[11] The final score of each corpus article is the sum of BOW and LSA scores and is assigned to the article URL. Afterwards, we run all the recommenders and intersect their result sets with the BOW+LSA ranking of the whole corpus, thus producing a so-called *semantic* ranking. This represents our final output, which consists of a ranked set of article URLs associated to the corresponding recommenders names.

[11] http://hlt.fbk.eu/en/technology/jlsi

6.4 Evaluation

The assessment of end user satisfaction has high priority in our work. According to Hayes et al. [58], we consequently decided to adopt an online evaluation approach with real users. In this scenario, the major issue consists of gathering a sufficiently large group of people who are willing to evaluate our systems. Crowdsourcing services provide a solution to the problem, as they allow us to outsource the evaluation task to an already available massive community of paid workers. To the best of our knowledge, no news recommender systems have been evaluated with crowdsourcing services so far. We set up an experimental evaluation framework for AMT, via the CrowdFlower platform.[12] A description of the mechanisms that regulate AMT is beyond the scope of the present paper: the reader may refer to [78] for a detailed analysis.

Our primary aim is to demonstrate that evaluators generally prefer our recommendations. Thus, we need to put our strategy in competition with a baseline. We leveraged the already implemented BOW+LSA information retrieval ranking algorithm. In addition, we set two specific objectives, related to the *specific explanation* and *unexpectedness* assumptions, as outlined in Section 6.1: (a) confirm that a specific explanation better attracts user attention rather than a generic one; (b) check if the recommended items are interesting, although they may appear unrelated and no matter what kind of explanation is provided.

Quality control of the collected judgements is a key factor for the success of the experiments. The essential drawback of crowdsourcing services relies on the cheating risk: workers (from now on called *turkers*) are generally paid a few cents for tasks which may only need a single click to be completed. Hence, it is highly probable to collect data coming from random choices

[12] http://crowdflower.com/

that can heavily pollute the results. The issue is resolved by adding *gold* units, namely data for which the requester already knows the answer. If a turker misses too many gold answers within a given threshold, he or she will be flagged as untrusted and his or her judgments will be automatically discarded.

6.4.1 General Setting

Our evaluation framework is designed as follows: (a) the turker is invited to read a complete news article. (b) A set of recommender systems are displayed below the article. Each system consists of a natural language *explanation* and a news title *recommendation*. (c) The turker is asked to give a preference on the most attracting recommendation, namely the one he or she would click on in order to read the suggested article. A single experiment (or *job*) is composed of multiple data *units*. A unit contains the text of the article and the set of explanation-recommendation pairs. Figure 6.2 shows a unit fragment of the actual web page that is given to a turker who accepted one of our evaluation jobs. Both instructions and question texts need to be carefully modeled, as they must mirror the main objective of the task and should not bias turkers' reaction. Since we aim at evaluating user attention attraction, we formulated them as per Figure 6.2.

6.4.2 Experiments

Table 6.2 provides an overview of our experimental environment. The parameters we have isolated for a single experiment are presented in Table 6.2a. On top of the possible variations, we built a set of nine experiments, which are described in Table 6.2b. We modeled two Q values, namely direct (as per Figure 6.2) and indirect (`Which recommendation do you consider to be more trustworthy?`), to monitor a possible alteration of turkers'

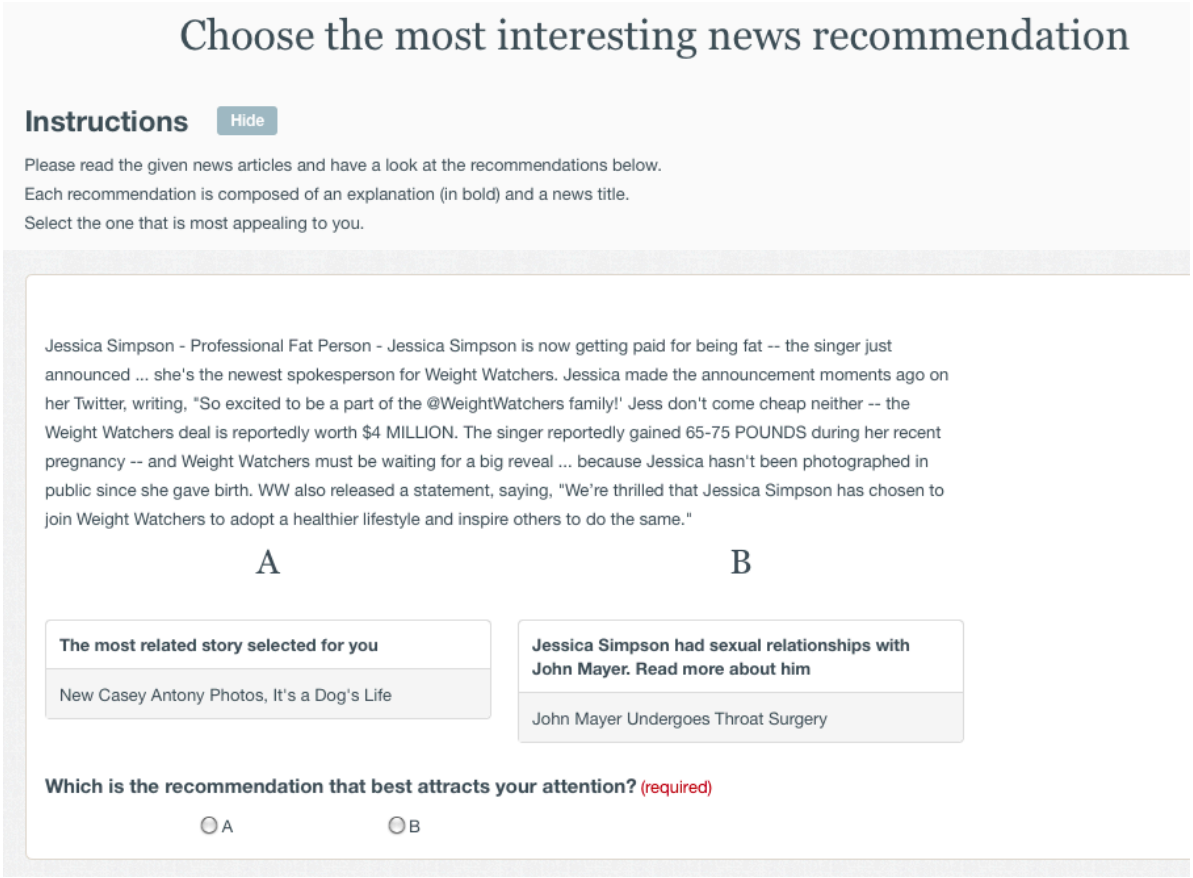

Figure 6.2: Web interface of an evaluation job unit

Table 6.2: Experiments overview

(a) Parameters

Parameter	Values
Q	D, I
A	2, 5
Exp	GS, G
SExp	SRO, SR, RO, R
Rec	B, S, F

(b) Configuration

Name	Q	A	Exp	SExp	Rec
Pilot	D	2	GS	SRO	B, S
Same explanation	D	2	G	None	B, S
4 generic + 1 specific	D	5	GS	SRO	B, S, F
5 generic	D	5	G	None	B, S, F
Same recommendation	D	2	GS	SRO	S
Relation only	D	5	GS	R	B, S, F
Subject + relation	D	5	GS	SR	B, S, F
Object + relation	D	5	GS	RO	B, S, F
Indirect	I	2	GS	SRO	B, S

Legend

Q	Question
A	Answer
Exp	Explanation
SExp	specific explanation
Rec	Recommendation

D	Direct
I	Indirect
2	Binary
5	5 choices
GS	Generic + specific
G	Generic only

SRO	Subject + relation + object
SR	Subject + relation
RO	Relation + object
R	Relation only
B	Baseline
S	Semantic
F	Fake

reaction. Experiments having $A = 5$ aim at decreasing the probability a turker gets trusted by chance, because he or she accidentally selected correct gold answers. They have an additional F value in the *Rec* parameter, as we randomly extracted 3 fake recommendations per unit from a file with more than 2 million news titles. However, such an architectural choice generated noisy results, since it occurred that some fake titles were selected.[13] *Exp* is a key parameter, which allows us to check whether the presence or the absence of a specific explanation represents a discriminating factor. *SExp* is intended to measure the effectiveness of a specific explanation while reducing its complexity.

Each job contains 8 regular + 2 gold units, namely 5 articles proposed twice, in combination with 2 significant (and eventually 3 fake) explanation-

[13]See Table 6.3 for further details.

recommendation pairs. The recommendation titles of the regular units are extracted from the top-2 links of the baseline and the semantic rankings. Gold is created by extracting the title from the last, i.e., less related link of the baseline ranking, the top link of the semantic ranking and assigning the correct answer to the latter. We collected a minimum of 10 valid judgments per unit and set the number of units per page to 3.

Once the results obtained, it frequently occurred that the expected number of judgments was higher: depending on their accuracy in providing answers to gold units, turkers switched from untrusted to trusted, thus adding free extra judgments. The proposed articles come from the TMZ website, which is well known in the United States. Therefore, we decided to gather evaluation data only from American turkers. The total cost of each experiment was 3.66$.

After visiting some news web portals, we chose the following generic explanations and randomly assigned them to both the baseline and the fake recommendations: (a) `The most related story selected for you`; (b) `If you liked this article, you may also like`; (c) `Here for you the hottest story from a similar topic`; (d) `More on this story`; (e) `People who read this article, also read`. 2 regular units were removed from the *relation only* and the *object + relation* experiments: it was impossible to build specific explanations with an implicit subject or object, since the entities that triggered the recommendations differed from the main entity of the source article.

6.4.3 Results

Table 6.3 provides an aggregated view of the results obtained from the Crowdflower platform.[14] With respect to the absolute percentage values, we

[14]The complete set of full reports is available at http://bit.ly/MOrN30

Table 6.3: Absolute results per experiment. ♢, ♠ and ♣ respectively indicate statistical significance differences between baseline and semantic methods, with $p < 0.05$, $p < 0.01$ and $p < 0.001$

Experiment	Judgments	Fake %	Baseline %	Semantic %
Pilot 1	82	0	40.24	**59.76**$^{\diamondsuit}$
Pilot 2	80	0	32.5	**67.5**$^{\spadesuit}$
Same explanation	80	0	48.75	51.25
4 generic + 1 specific	90	3.33	23.33	**73.33**$^{\clubsuit}$
5 generic	88	13.63	37.5	48.86
Same recommendation	86	0	36.04	**63.96**$^{\spadesuit}$
Relation only	68	13.23	41.17	45.58
Indirect	82	0	37.8	**62.2**$^{\spadesuit}$
Subject + relation	86	8.13	41.86	50
Object + relation	68	5.88	41.17	52.94

first observe that our approach always outperformed the baseline. Furthermore, statistical significance differences emerge when a complete $<s, r, o>$ specific explanation is given. We ran twice, i.e., in two separate days the *pilot* experiment and noticed an improvement. The *indirect* experiment only differs from the pilot in the question parameter and yielded similar results. The *4 generic + 1 specific* experiment has the highest semantic percentage: this translates into an expected behavior, since the presence of a single specific explanation against four generic ones is likely to bias turkers' reaction towards our approach. As the complexity of the specific explanation decreases, i.e., in the *subject + relation, object + relation* and *relation only* experiments or when only generic explanations are presented, namely in the *5 generic* and *same explanation* experiments, judgments towards our approach tend to decrease too. Hence, we evince the importance of providing specific explanations in order to attract user attention.

6.4.4 Discussion

Experiments containing a specific explanation aim at assessing its attractive power (assumption 3). If we compare experiments which only differ in the *Exp* parameter, namely *4 generic + 1 specific* and *5 generic*, *pilot 1-2* and *Same explanation*, in the formers turkers prefer our strategy with a statistically significant difference. Therefore, specific explanations are proven to enhance the trustworthiness of the system.

The evaluation of the unexpectedness factor (assumption 2) boils down to check whether turkers privilege the novelty of a recommendation or its similarity to the source article. In experiments including only generic explanations, namely *Same explanation* and *5 generic*, we noticed the following: (a) no statistically significant differences exist between the strategies; (b) when the baseline returns articles that are unrelated to the topic or the entity of the source article, turkers prefer our strategy and vice versa. Hence, we argue that users tend to privilege similarity if they are given a generic explanation. On the other hand, when the baseline strategy suggests a clearly related article and when a specific explanation is provided, turkers tend to choose our strategy even if it suggests an apparently unrelated article. This is a first proof of the unexpectedness factor: users are attracted by the specific explanation and are eager to read an unexpected article rather than another article on the same topic/entity.

6.5 Conclusion

In this chapter, we presented a novel recommendation strategy leveraging entity linking techniques in unstructured text and knowledge extraction from structured knowledge bases. On top of it, we build hybrid entity-oriented recommender systems for news filtering and post-click news recommendation. We argued that entity relations discovery leads to unexpected suggestions

and specific explanations, thus attracting user attention. The adopted online evaluation approach via crowdsourcing services assessed the validity of our systems. A demo prototype consumes Freebase data to recommend TMZ celebrity gossip articles.

Chapter 7

Conclusion

In a continuously evolving World Wide Web (WWW), where machines stand beside humans as content consumers, there is an ever growing need for shaping information in such a way that it can be understood by both. In this context, *Knowledge Bases* (KBs) play a critical role: they supply structured facts about diverse domains and attempt to consistently store them in formal classification schemata, or *ontologies*. In a world burdened by information overload, the paradigm would allow the construction of intelligent automated agents, which can interpret the WWW content and directly satisfy the needs of human users. This is already becoming a reality, as the largest Web companies have adopted KB-driven solutions to power their platforms, the most renowned being Google's Knowledge Graph,[1] Facebook's Graph Search,[2] and Microsoft's Satori.[3] Furthermore, KBs dispense the fuel to run a wide range of applications, from cognitive computing systems like IBM Watson[4] to knowledge engines such as Wolfram Alpha[5], all the way to personal assistants, e.g., Apple's Siri[6].

[1] https://www.google.com/intl/en_us/insidesearch/features/search/knowledge.html
[2] http://www.facebook.com/about/graphsearch
[3] https://blogs.bing.com/search/2015/08/20/bing-announces-availability-of-the-knowledge-and-action-graph-api-for-developers/
[4] http://www.ibm.com/smarterplanet/us/en/ibmwatson/index.html
[5] http://www.wolframalpha.com/
[6] http://www.apple.com/ios/siri/

However, all the aforementioned systems require high-quality data in order to deliver the best answers and to avoid wrong ones. State-of-the-art KBs are proven to leverage a massive amount of (probably tedious) manual curation labor: for instance, FREEBASE has built an entire human computation engine [69] to augment the value of its data; WIKIDATA [125] is itself conceived as a fully collaboratively edited resource, following the Wiki fashion. To a certain ironic extent, this sounds like "Artificial artificial intelligence", just to cite the motto of a notable *crowdsourcing* platform, i.e., Amazon's MECHANICAL TURK.[7]

While we argue that human intervention cannot be completely eliminated, we concentrate on minimizing its necessity. Therefore, we focus on DBPEDIA [73], a KB which is still devoted to the development of an automatic extraction framework from Wikipedia content. The core purpose of this thesis is to **improve the DBpedia data quality** by means of Natural Language Processing (NLP) techniques, which aim at reducing manual curation. For that reason, we investigated issues related to the DBpedia classification system, i.e., the ontology (DBPO). More specifically, we addressed the problems highlighted in Section 1.2, namely its **unbalanced nature** and the **lack of coverage**. The results of our research are incorporated as a tangible proof of work in the *Italian DBpedia chapter* and are summarized in the following sections, in which we include specific prospects for future work.

StrepHit: a Project Funded by the Wikimedia Foundation Besides DBpedia, we have also concentrated our latest efforts to improve the data quality of Wikidata. We would like to recall that our STREPHIT project proposal won the **largest Wikimedia Foundation Individual Engagement Grant**, 2015 round 2 call. This allowed us to conduct further

[7] https://www.mturk.com/mturk/welcome

research based on Contribution 2 (cf. Section 7.3), under the umbrella of one of the most wide-reaching non-governative organizations in the world. StrepHit originates from the efforts described in Chapter 4 and prosecutes the vision of Intelligent Agents: specifically, it targets the creation of a Web-reading Agent that would validate Wikidata content via facts extracted from third-party Web sources. The selected project proposal is available at https://meta.wikimedia.org/wiki/Grants:IEG/StrepHit:_Wikidata_Statements_Validation_via_References.

7.1 The Italian DBpedia Chapter

The core practical outcome of this work is the foundation, the development, and the maintenance of the Italian DBpedia chapter, which is online at http://it.dbpedia.org. We underline once again the role of the author as a (1) member of the DBpedia Association board of trustees,[8] and an (2) organization administrator for the Google Summer of Code program, thus also providing financial resources to the Association.

Back to its foundation in 2012, which has also been possible thanks to the collaboration with the startup SpazioDati[9], we detail below the principal achievements of the current Italian DBpedia chapter deployment.

1. **Best Dataset Award** at Apps4Italy: we won the first prize in the 'datasets' category at the Apps4Italy competition. This achievement was reported on La Repubblica newspaper[10] and informally resumed in the following blog post: http://it.dbpedia.org/2012/05/dbpedia-italiana-premiata-apps4italy/ (in Italian);

[8] https://docs.google.com/document/d/1pchRPLtQwO3GH49cF7GB33srRn1p2M3QW8Jtu5b5ZwE/edit

[9] https://spaziodati.eu

[10] http://www.repubblica.it/tecnologia/2012/05/19/news/premio_app4italy-35459551/

2. the **Mapping Sprint**: we first conducted a teaching activity via the FBK JUNIOR program[11] and trained a team of high school students to the DBpedia ontology (DBPO) manual mapping workflow.[12] Then, we organized and held a hackathon at the BERTRAND RUSSELL high school,[13] where the team members served as mentors. This achievement was informally resumed in the following blog post: http://it.dbpedia.org/2014/04/grande-successo-per-il-primo-mapping-sprint/?lang=en;

3. AIRPEDIA **classes integration**: the automatic DBPO classes mapping approach presented in [4] is integrated. This achievement was informally resumed in the following blog post: http://it.dbpedia.org/2013/06/nuova-release-dbpedia-3-2-airpedia-wikidata/?lang=en;

4. LODVIEW **data visualization**: LODVIEW[14] becomes the official data visualization tool. This achievement was informally resumed in the following blog post: http://it.dbpedia.org/2015/06/lodview-nuova-veste-grafica-per-i-dati-della-dbpedia-italiana-2/?lang=en;

5. the **Italian soccer dataset**: the results of Contribution 2 (Chapter 5), part of the Google Summer of Code 2015 project "Fact Extraction from Wikipedia", are integrated. This achievement was informally resumed in the following blog post: http://it.dbpedia.org/2015/09/meno-chiacchiere-piu-fatti-una-marea-di-nuovi-dati-estratti-dal-testo-di-wikipedia/?lang=en;

6. DBTAX **integration**: the results of Contribution 3 (Chapter 5), part

[11] http://airt.fbk.eu/it/relazioni-con-le-giovani-generazioni
[12] http://mappings.dbpedia.org
[13] http://www.liceorussell.eu
[14] http://lodview.it/

of the Google Summer of Code 2013 project "Type Inference to Extend Coverage", are integrated. This achievement was informally resumed in the following blog post: http://it.dbpedia.org/2015/02/dbpedia-italiana-release-3-4-wikidata-e-dbtax/?lang=en;

7. **stakeholders**: we list below those entities that have explicitly stated their use of the Italian DBpedia chapter.

   - Digital libraries:
     - University of Urbino digital library;[15]
     - National Central Library of Florence, Nuovo Soggettario Thesaurus.[16]
   - Data-driven journalism:
     - Focus magazine, article "Le misure del calcio";[17]
     - Inchiesta journal, dossier on Expo "Milano, oggi domani dopodomani".[18]

7.2 Contribution 1: Crowdsourced Frame Annotation

In Chapter 3, we proposed a methodology to perform full Frame Semantics annotation in a crowdsourcing environment. Our core research contribution relies in the reduction to a *single-step, bottom-up* task, as opposed to the usual workflow consisting of two steps, one for *frame* disambiguation, and one for Frame Elements (*FEs*) assignment. The former is intrinsically bound to the latter, since it can be fulfilled only if annotators decide on the

[15] http://opac.uniurb.it/SebinaOpac/.do#0
[16] http://thes.bncf.firenze.sbn.it/
[17] http://www.focus.it/temi/le-misure-del-calcio
[18] http://www.inchiestaonline.it/archivio/e-uscito-il-numero-188-di-inchiesta-aprile-giugno-2015/, http://milano-odd.it/?p=594

frame by implicitly matching its FEs with the participants represented in the sentence. This implies that the cognitive process already involves the identification of FEs in the first step, even if they are explicitly labeled only in the second step. Consequently, we also claim that our novel annotation approach better adheres to the original linguistic theory illustrated in [47].

We carried out a set of experiments via the CrowdFlower platform, following two strategies: one with manually simplified FE definitions based on the FrameNet resource, and one with automatically derived suggestions leveraging DBPO class labels. The collected results first demonstrate that our 1-step approach is not only cheaper than the 2-step one in terms of execution time (**-24%**), but also yields more accurate annotations (**+15%**), although at a higher financial cost (**+84%**), due to the higher number of questions that need to be asked. Moreover, we completely substitute the confusion-prone FE definitions with automatic hints extracted from DBpedia, and further improve both the overall annotation accuracy (+11.4%) while dramatically decreasing the execution time (-51%).

Future work will include the refinement of the frame assignment strategy. In fact, we do not take into account the case of conflicting FE annotations in cross-frame units. Hence, we need a confidence score to determine which frame emerges if workers selected contradictory answers in a subset of cross-frame FE definitions. Secondly, the evaluation of an ad-hoc strategy for the extraction of semantic types is needed, in order to provide workers with suggestions that are dynamically derived from each given sentence rather than statistics. Furthermore, clustering of similar semantic types with respect to the meaning they convey and to the frequency, e.g. `Place` and `Location_Underspecified`. Finally, the overall effectiveness of our approach depends both on the performance of the entity linking system and on the coverage of the knowledge base. Hence, long term research will focus on enhancing The Wiki Machine precision and recall, and extending

DBpedia type coverage.

7.3 Contribution 2: Properties Population via Relation Extraction

In Chapter 4, we designed a Relation Extraction pipeline that implements the crowdsourcing methodology as per **Contribution 1** and enriches our target KB DBpedia with properties extracted from Wikipedia free text. We are set apart from related state-of-the-art systems with respect to four features:

1. *N*-ary Relation Extraction enabled by Frame Semantics, whereas standard approaches are binary;
2. simultaneous T-Box (i.e., new properties) and A-Box (i.e., new assertions) DBpedia augmentation;
3. economical NLP technology, requiring POS-tagging only, instead of more complex layers, e.g., constituency parsing;
4. completely supervised yet low-cost learning paradigm thanks to crowdsourcing, in contrast with noisy unsupervised or distantly supervised techniques.

We assessed the effectiveness of our system through the soccer use case in Italian. Given as input circa 52,000 Italian Wikipedia articles describing soccer players, we produced a dataset of more than 210,000 triples with an average performance of 78.5% F1. We estimated the target KB (i.e., the Italian DBpedia chapter) coverage improvement in two ways: first, we calculated a relative gain of **+96.8%** new confident A-Box assertions with respect to pre-existing ones. In addition, 50% frames and 80% FEs of the use case frame set represent novel T-Box properties. It is clear that these

T-Box results are promising, although we recognize that the size of the analysed frame set is still too small for a statistical significance validation.

The project has been carried out via the Google Summer of Code 2015 program,[19] under the umbrella of the DBpedia Association: its codebase is available at https://github.com/dbpedia/fact-extractor and has attracted considerable interest in the open source landscape.

For general future work, we foresee to scale up the implementation towards multilingual open information extraction, thus paving the way to (a) its full deployment into the DBpedia Extraction Framework, and to (b) a thorough referencing system for Wikidata.

We summarize below some fine-grained technical aspects that can be considered as improvements, inviting the reader to refer to Chapter 4 for a detailed description. We handled Lexical Units (LUs) consisting of one single token (unigrams), but we could attain more recall if we considered n-grams. Tagging n-grams with DBPO classes retrieved during the entity linking step may be an impactful additional feature to train the FE classifier. The gazetteer is currently run at the token level, but it may be more useful to run it over the whole input (i.e., sentence). In order to reduce the noise in the training set, we foresee to leverage a sentence splitter and extract 1-sentence examples only. Further less strict evaluation experiments will take into account the classified n-grams instead of disambiguated links. Include the frame confidence for further refinement of the final confidence score.

[19] http://www.google-melange.com/gsoc/homepage/google/gsoc2015

7.4 Contribution 3: Classes Population via Taxonomy Learning

In Chapter 5, we illustrated a fully data-driven procedure to learn a wide-coverage general-purpose taxonomy from the Wikipedia category system, and to employ it for jointly enriching the DBpedia T-Box and A-Box with respect to classes and *instance-of* assertions.

We estimated a remarkable coverage improvement (**+93.7%**) compared to the current DBpedia main classification system (i.e., DBPO), with **4.2 million** versus only 2.2 million typed resources respectively. While we acknowledge that a considerable amount of related work has been conducted prior to ours, we argue that no focus has been accorded so far to (a) the actual usability of the resource, and (b) its integration into a well-established framework such as DBpedia. Therefore, we executed online crowdsourced evaluations with real-world non-expert users, which prove that our system is not only equivalent to the handcrafted DBPO with regards to its structure, but is also distinctively intuitive, while still outperforming automatic analogous efforts in terms of precision and recall trade-off.

For future work, we plan to merge the T-Box into the DBpedia mappings wiki[20] and allow the DBpedia community to further curate and organize it. We believe this will also cater for the broad hierarchy paths that resulted from the pruning steps. Furthermore, a word sense disambiguation technique is scheduled for implementation, in order to distinguish between homonymous classes. Since the A-Box may state multiple heterogeneous types for a resource (e.g., Elvis Presley is both a Singer and a Protestant), we foresee to rank types according to their statistical relevance, such as the absolute frequency of instances. Finally, we expect to additionally

[20] http://mappings.dbpedia.org

exploit the Wikipedia category interlanguage links, in order to (a) produce multilingual labels for DBTax, (b) pinpoint additional classes that our process did not extract in English, and (c) deploy the approach to DBpedia language chapters besides English and Italian, at the price of excluding categories with no English counterpart.

7.5 Contribution 4: Application to Recommender Systems

In Chapter 6, we developed an innovative recommendation method that cooperatively exploits a KB and entity linking in order to deliver unusual, hence serendipitous, suggestions. A news recommender system is constructed upon it, which we believe to serve as an end-user application that demonstrates the potential use of KBs in a real-world setting. The engine is a hybrid between content-based and knowledge-based approaches: the former transforms an input corpus of documents into a structured dataset that integrates into the target KB as a fused queryable dataspace. Therefore, the discovery of relations between its entities enables both the delivery of unexpected recommendations and the generation of detailed explanations, thus being attractive to end users. We performed several online crowdsourced evaluation experiments that demonstrate the benefits of our strategy compared to a baseline, and are further supported by statistical significance. A use case prototype consumes data from Freebase and recommends TMZ[21] celebrity gossip news articles.

For our future work, we have set the following milestones. (a) *Comparison with DBpedia*: our use case leverages an off-the-shelf KB, which is heavily curated by hand. Hence, we plan to perform a comparative analysis by switching to a version of DBpedia that is automatically populated by

[21] http://www.tmz.com

our techniques. (b) *Ecological evaluation*: for the online evaluation, we used the CrowdFlower platform, which allowed us to build fast and cheap experiments. However, the collected judgments may be biased by the politeness effect of the economical reward and the turkers' awareness of performing a question-answering task. Therefore, we intend to set up an ecological evaluation scenario, which simulates a fully real-world usage of our recommender systems and enables natural user reactions. We foresee to adopt the Google AdWords[22] approach proposed in [56]. (c) *Methodology for building recommenders*: currently, we have manually implemented a domain-specific list of recommenders, based on the most frequent corpus concepts. We plan to automate this process by extracting generic relations from Freebase via data analytics techniques. (d) *Methodology for building specific explanations*: explanations are naively mapped to the relations and the corresponding subject/object entities. How to automatically build linguistically correct sentences remains an open problem. (e) *User profile construction*: explicit and implicit user preferences acquisition can improve the quality of the recommendations.

[22] http://adwords.google.com/

Chapter 8

Appendix: the StrepHit Project

This appendix contains the technical reports of StrepHit, the project funded by the Wikimedia Foundation through an Individual Engagement Grant (IEG).

8.1 Project Idea

StrepHit (pronounced "*strep hit*", means "*Statement? repherence it!*") is a Natural Language Processing pipeline that harvests structured data from raw text and produces Wikidata statements with reference URLs. Its datasets will feed the Primary Sources tool.[1] In this way, we believe StrepHit will dramatically improve the data quality of Wikidata through a reference suggestion mechanism for statement validation, and will help Wikidata become the gold-standard hub of the Open Data landscape.

8.1.1 The Problem

The trustworthiness of Wikidata assertions plays the most crucial role in delivering a high-quality, reliable Knowledge Base: in order to assess their truth, assertions should be validated against third-party resources, and few

[1] https://www.wikidata.org/wiki/Wikidata:Primary_sources_tool

efforts have been carried out under this perspective. One form of validation can be achieved via references to external (i.e, non-wiki), authoritative sources. This has motivated the development of the primary sources tool: it will serve as a platform for users to either accept or reject new references and/or assertions coming from third-party datasets. We argue that there is a need for datasets which guarantee at least one reference for each assertion, and StrepHit is conceived to do so.

8.1.2 The Solution

StrepHit applies Natural Language Processing techniques to a selected corpus of authoritative Web sources in order to harvest structured facts. These will serve two purposes: to authenticate existing Wikidata statements, and ultimately to enrich them with references to such sources. More specifically, the solution is based on the following main steps:

1. Corpus-based relation discovery, as a completely data-driven approach to knowledge harvesting;
2. Linguistically-oriented fact extraction from reliable third-party Web sources.

The solution details are best explained through the use case shown below.

8.1.3 Use Case

Soccer is a widely attested domain in Wikidata: it counts a total of 188,085 items describing soccer-related entities,[2] which is a significant portion

[2] According to the following query: http://tools.wmflabs.org/autolist/autolist1.html?q=claim{[}31:(tree{[}1478437{]}{[}{]}{[}279{]}){]}%20or%20claim{[}31:(tree{[}15991303{]}{[}{]}{[}279{]}){]}%20or%20claim{[}31:(tree{[}18543742{]}{[}{]}{[}279{]}){]}%20or%20claim{[}106:628099{]}%20or%20claim{[}106:937857{]}

(around 1.27%) of the whole knowledge base. Moreover, those Items are generally very rich in terms of statements (cf. for instance the Germany national football team).[3]

On account of such observations, the soccer domain properly fits the main challenge of this proposal, namely to automatically validate Wikidata statements against a knowledge base built upon the text of third-party Web sources (from now on, the WEB SOURCES KNOWLEDGE BASE).

The following list displays four example statements with no reference from the *Germany national football team* Wikidata Item, which can be validated by candidate statements extracted from the given references.

1. (Germany, participant of, Miracle of Cordoba)
   - The Telegraph[4]
   - *"(...) The Miracle of Cordoba, when they eliminated Germany from the 1978 World Cup"*
   - (Germany, eliminated in, Miracle of Cordoba)
2. (Germany, team manager, Franz Beckenbauer)
   - Encyclopædia Britannica[5]
   - *"In 1984 Beckenbauer was appointed manager of the West German team"*
   - (West German team, manager, Beckenbauer)
3. (Germany, inception, 1908)
   - DFB[6]

[3] https://www.wikidata.org/wiki/Q43310

[4] http://www.telegraph.co.uk/sport/football/international/2304101/Euro-2008-Germany-end-Turkeys-fairytale.html

[5] http://www.britannica.com/biography/Franz-Beckenbauer

[6] http://www.dfb.de/en/national-teams/

- *"The story of the DFB's national team began (...) on April 5th 1908"*
- (DFB's national team, start, 1908)

4. (Germany, captain, Michael Ballack)
   - Spiegel[7]
   - *"Michael Ballack, the captain of the German national football team"*
   - (German national football team, captain, Michael Ballack)

Proof of Work

The soccer use case has already been partially implemented: the prototype has yielded a small demonstrative dataset, namely STREPHIT-SOCCER, which has been uploaded to the primary sources tool.

8.2 Project Goals

The technical goals of this project are as follows:

1. to identify a set of authoritative third-party Web sources and to harvest the *Web Sources Corpus*;
2. to recognize important *relations* between entities in the corpus via lexicographical and statistical analysis;
3. to implement the *StrepHit* Natural Language Processing pipeline, serving in all respects as an *open source framework* that maximizes reusability;

[7] http://www.spiegel.de/international/germany/ankle-injury-german-team-captain-michael-ballack-ruled-out-of-world-cup-a-695164.html

4. to build the *Web Sources Knowledge Base* for the validation and enrichment of Wikidata statements;

5. to deploy a stable system that automatically suggests references given a Wikidata statement.

Community Outreach

The target audience is represented by several communities: each one will play a key role at different phases of the project, and will be attracted accordingly. We list them below, in descending order of specificity:

- Wikidata users, involved as data curators;
- Wikipedia users and librarians, involved as consultants for the identification of reliable Web sources;[8]
- technical contributors (i.e., Natural Language Processing developers and researchers), involved through standard open source and social coding practices;
- data donors, encouraged by the availability of a unified platform to push their datasets into Wikidata.

8.3 Project Plan

8.3.1 Implementation Details

We scale up the approach described in Chapter 4: we take as input a collection of documents from a set of Web sources (i.e., the *corpus*) and

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

output a structured knowledge base composed of machine-readable *statements* (according to the Wikibase data model terminology).[9] The workflow is depicted in Figure 8.1.

8.3.2 Contributions to the Wikidata Development Plan

In general, this project is intended to play a central role in the primary sources tool. A list of specific open issues follows.

1. Framework for source checking, T90881:[10] StrepHit seems like a perfect match for this issue;
2. Nudge editors to add a reference when adding a new claim, T76231:[11] Automatically suggesting references would encourage editors to fulfill these duties;
3. Nudge when editing a statement to check reference, T76232:[12] same as above.

8.3.3 Work Package

The work package consists of the following tasks:

1. Development corpus: gather 200,000 documents from 40 authoritative Web sources;
2. State of the art review: investigate reusable implementations for the StrepHit pipeline;
3. Corpus analysis: select the top 50 verbal lexical units that emerge from the corpus;

[9] https://www.mediawiki.org/wiki/Wikibase/DataModel
[10] https://phabricator.wikimedia.org/T90881
[11] https://phabricator.wikimedia.org/T76231
[12] https://phabricator.wikimedia.org/T76232

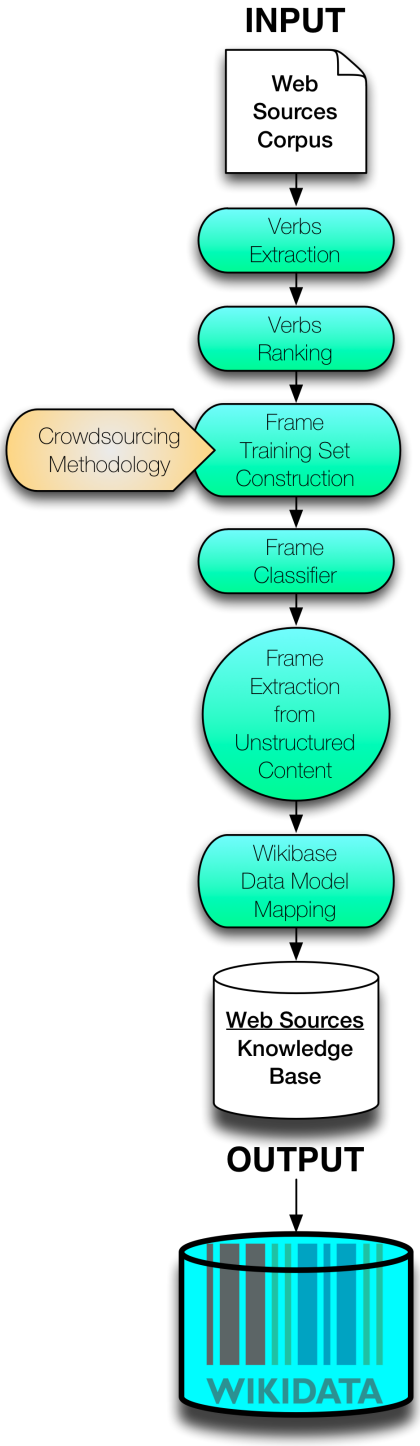

Figure 8.1: StrepHit workflow

4. Production corpus: regularly harvest 50,000 new documents from the selected sources;
5. Training set: construct the training data via crowdsourcing;
6. Classifier testing: train and evaluate the supervised classifier to achieve reasonable performance;
7. Frame extraction: transform candidate sentences of the input corpus into structured data via frame classification;
8. Web Sources Knowledge Base: produce the final 2.25 million statements dataset and upload it to the primary sources tool;
9. Stable primary sources tool: fix critical issues in the codebase;
10. Community dissemination: promote the project and engage its key stakeholders.

8.4 Community Engagement

All the following target communities have been notified before the start of the project and will be involved according to the different phases:

- Wikidatans;
- Wikipedians;
- Librarians (and GLAM-related[13] communities);
- Natural Language Processing developers and researchers;
- Open Data organizations.

[13] https://en.wikipedia.org/wiki/Wikipedia:GLAM

The engagement process will mainly be based on a constant presence on community endpoints and social media, as well as on the physical presence of the project leader to key events.

Phase 0: Testing the Prototype. The STREPHIT-SOCCER demonstrative dataset contains references extracted from sources in Italian. Hence, we have invited the relevant Italian communities to test it.

Phase 1: Corpus Collection. The Wikipedia community has defined comprehensive guidelines for sources verifiability.[14] Therefore, it will be crucial to the early stage of the project, as it can discover and/or review the set of authoritative Web sources that will form the input corpus. Librarians are also naturally vital to this phase, due to the relatedness of their work activity.

Phase 2: Multilingual StrepHit. Besides the Italian demo dataset, the first StrepHit release will support the English language. We aim at attracting Natural Language Processing experts to implement further language modules, since Wikidata publishes multilingual content and benefits from a multilingual community. We believe that references from sources in multiple languages will have a huge impact in improving the overall data quality.

Phase 3: Further Data Donation. The project outcomes will serve as an encouragement for third-party Open Data organizations to donate their data to Wikidata through a standard workflow, leveraging the primary sources tool.

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

8.5 Methods and activities

8.5.1 Technical Setup

We requested the credentials and created a GitHub repository within the Wikidata organization.[15] The official documentation page is hosted at `mediawiki.org`.[16] Besides the planned work package, special development efforts have been devoted to:

- a modular architecture;
- parallel processing;
- caching;
- let StrepHit be used both as a library and as a set of command line tools;
- an easy-to-use command line to run all the pipeline steps;
- a flexible logging facility.

8.5.2 Project Management

The project has undergone the following activities:

- Monday face-to-face meetings for brainstorming ideas and weekly planning;
- daily scrums, especially for unexpected technical issues, but also for brainstorming;
- whiteboard for crystallized ideas;
- yellow stickers on the whiteboard for ideas to be investigated;

[15] https://github.com/Wikidata/StrepHit
[16] https://www.mediawiki.org/wiki/StrepHit

- regular interaction with relevant mailing lists and key people to discuss potential impacts and to gather suggestions;
- project dissemination in the form of seminars and talks.

8.5.3 Dissemination

We conducted the following dissemination efforts.

- Kick-off seminar
  - Video: https://www.youtube.com/watch?v=uvfd_HmPOrc
  - Slides: http://www.slideshare.net/MarcoFossati/strephit-ieg-kickoff-seminar
- Event at Lugano: http://www.ated.ch/manifestazioni/7/web-30-il-potenziale-del-web-semantico-e-dei-dati-strutturati_3194.html (in Italian)
- HackAtoka hackathon: http://blog.atoka.io/hackatoka-open-innovation-al-lavoro-per-testare-le-nuove-atoka-api/ (in Italian)
- Spaghetti Open Data Reunion hackathon: http://www.spaghettiopendata.org/content/wikidata-la-banca-di-conoscenza-libera-casa-wikimedia
- WikiCite 2016:
  - Main page: https://meta.wikimedia.org/wiki/WikiCite_2016
  - Proposal: https://meta.wikimedia.org/wiki/WikiCite_2016/Proposals/Generation_of_referenced_Wikidata_statements_with_StrepHit
  - Work group: https://meta.wikimedia.org/wiki/WikiCite_2016/Report/Group_4

- Wikimania 2016 poster: https://wikimania2016.wikimedia.org/wiki/Posters#StrepHit
- Request for comment: https://www.wikidata.org/wiki/Wikidata:Requests_for_comment/Semi-automatic_Addition_of_References_to_Wikidata_Statements

8.6 Outcomes

The key planned outcomes of StrepHit are:

- the *Web Sources Corpus*, composed of 1.6 M items circa gathered from 53 reliable Web sources;
- the *Natural Language Processing pipeline* to extract Wikidata claims from free text;
- the *Web Sources Knowledge Base*, composed of 2.6 M Wikidata claims circa.

The following list illustrates the output produced by the StrepHit project.

1. Web Sources Corpus: 1,623,381 items, 504,189 documents, 53 sources;
2. Candidate Relations Set: 49 frames, 229 total frame elements, 133 unique frame elements, 69 unique Wikidata relations;
3. StrepHit Pipeline Beta: version 1.0 beta and 1.1 beta released;
4. Web Sources Knowledge Base: 842,208 confident, 958,491 supervised, 808,708 rule-based, 2,609,407 total Wikidata claims;
5. primary sources tool: 5 merged pull requests, active community discussion.

8.6.1 Software

The following modules have reached a mature state from a software development perspective:

- `web_sources_corpus`,[17] i.e., a set of Web spiders that harvest data from the selected biographical authoritative sources;[18]
- `corpus_analysis`,[19] i.e., a set of scripts to process the corpus and to generate a ranking of the candidate relations;
- `commons`,[20] i.e., several facilities to ensure a scalable and reusable codebase. On the general-purpose hand, these include parallel processing, fine-grained logging, and caching. On the specific Natural Language Processing (NLP) hand, special attention is paid to foster future multilingual implementations, thanks to the modularity of the NLP components, such as tokenization,[21] sentence splitting,[22] and part-of-speech tagging.[23]
- `extraction`,[24] i.e., the logic needed to extract different set of sentences, to be used for training and testing the classifier, as well as for the actual production of Wikidata content;
- `annotation`,[25] i.e, a set of scripts to interact with the CrowdFlower crowdsourcing platform APIs, in order to create and post annotation jobs, and to pull results.

[17] https://github.com/Wikidata/StrepHit/tree/master/strephit/web_sources_corpus
[18] https://github.com/Wikidata/StrepHit/issues/13
[19] https://github.com/Wikidata/StrepHit/tree/master/strephit/corpus_analysis
[20] https://github.com/Wikidata/StrepHit/tree/master/strephit/commons
[21] https://github.com/Wikidata/StrepHit/blob/master/strephit/commons/tokenize.py
[22] https://github.com/Wikidata/StrepHit/blob/master/strephit/commons/split_sentences.py
[23] https://github.com/Wikidata/StrepHit/blob/master/strephit/commons/pos_tag.py
[24] https://github.com/Wikidata/StrepHit/tree/master/strephit/extraction
[25] https://github.com/Wikidata/StrepHit/tree/master/strephit/annotation

8.6.2 Bonus Outcomes

Besides the planned goals, we reached the following bonus outcomes, in order of relevance to the Wikidata community:

1. the *unresolved entities* dataset. When generating the Web Sources Knowledge Base, a (rather large) set of entities could not be resolved to Wikidata QIDs. They may serve as candidates for new Wikidata Items;

2. the *Wiki Loves Monuments for Wikidata* prototype dataset. We were contacted by Wikimedia Italy to implement a very first integration of a WLM Italy dataset into Wikidata;

3. a *rule-based* statement extraction technique, which does not require any training set, although it may yield less accurate extractions. It can be thought as a trade-off between the text annotation and the statement validation costs;

4. the *Italian companies* dataset, as a result of the HackAtoka hackathon. It is a proof of scalability for the StrepHit pipeline: the rule-based technique has been succesfully applied to another domain (companies), in another language (Italian).

8.6.3 Web Sources Corpus

Table 8.1 displays raw counts of scraped items and biographies grouped by Web domains. Together they constitute the input corpus of this project. Since `en.wikisource.org` actually embeds several sources, Table 8.2 breaks them down. A considerable slice of the corpus does not contain any free text document, but rather semi-structured data that should be exploited in parallel to the NLP pipeline. This is reflected in Figure 8.3, showing the

distribution of items with biographies and without biographies. From a document length perspective, we observe a high density of short biographies, as depicted in Figure 8.2, which plots the distribution of biographies according to their length in characters. Figure 8.5 and Figure 8.4 respectively detail how items and biographies are distributed across sources.

Source domain	items	biographies
www.genealogics.org	447,045	10,621
www.metal-archives.com	355,784	7,988
rkd.nl	206,993	
vocab.getty.edu	199,502	199,496
collection.britishmuseum.org	118,883	101,117
en.wikisource.org	60,403	60,355
www.nndb.com	40,331	40,331
www.bbc.co.uk	38,018	1,321
www.catholic-hierarchy.org	37,313	
www.daao.org.au	19,696	9,848
adb.anu.edu.au	19,086	19,086
gameo.org	13,858	13,850
www.uni-stuttgart.de	10,679	
archive.org	8,721	8,719
cesar.org.uk	7,044	
munksroll.rcplondon.ac.uk	6,959	6,921
sculpture.gla.ac.uk	6,378	5,631
structurae.net	6,340	
yba.llgc.org.uk	4,470	4,470
www.wga.hu	3,952	3,927
collection.cooperhewitt.org	3,407	3,407
dictionaryofarthistorians.org	2,442	2,259

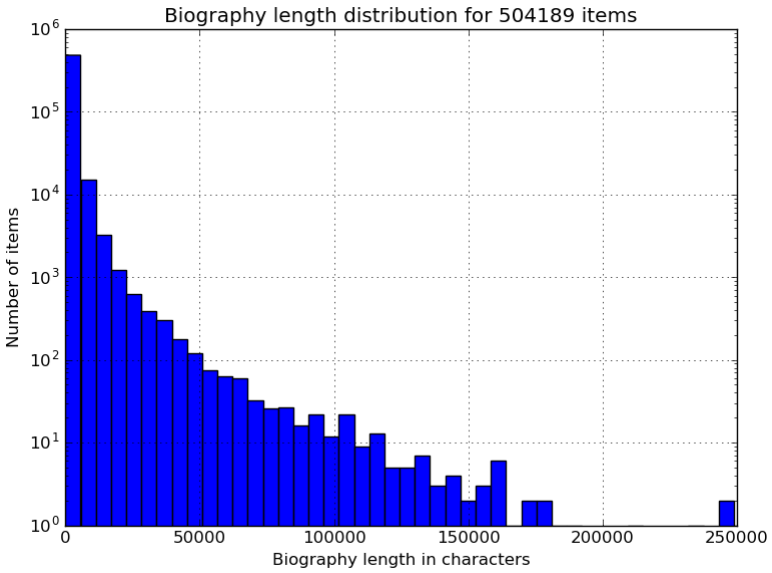

Figure 8.2: Distribution of StrepHit IEG Web Sources Corpus Biographies according to the length in characters

Source domain	items	biographies
www.newulsterbiography.co.uk	2,060	2,060
royalsociety.org	1,596	1,580
www.parliament.uk	650	
www.museothyssen.org	627	585
www.brown.edu	601	601
www.academia-net.org	525	
Total	**1,623,381**	**504,189**

Table 8.1: Items and biographies across Web domains

Source	items	biographies
DNB	28001	27997
Catholic Encyclopedia	11466	11462
Naval Bio	4692	4688

Source	items	biographies
Indian Bio	2440	2427
American Bio	2209	2207
National Bio 1912	1631	1631
Australasian Bio	1590	1590
Irish Officers	1530	1524
Bio English Lit	1346	1340
Men at the Bar	1115	1115
National Bio 1901	1033	1033
Christian Bio	921	921
Musicians	702	702
Freethinkers	546	546
Men of Time	432	431
Chinese Bio	245	245
English Artists	223	223
Medical Bio	109	109
Portraits and Sketches	50	50
Who is who in China	47	47
Greek Roman bio Myth	37	37
Modern English Bio	11	11
Who is who America	10	10
Total	**60,403**	**60,355**

Table 8.2: Items and biographies Wikisource breakdown

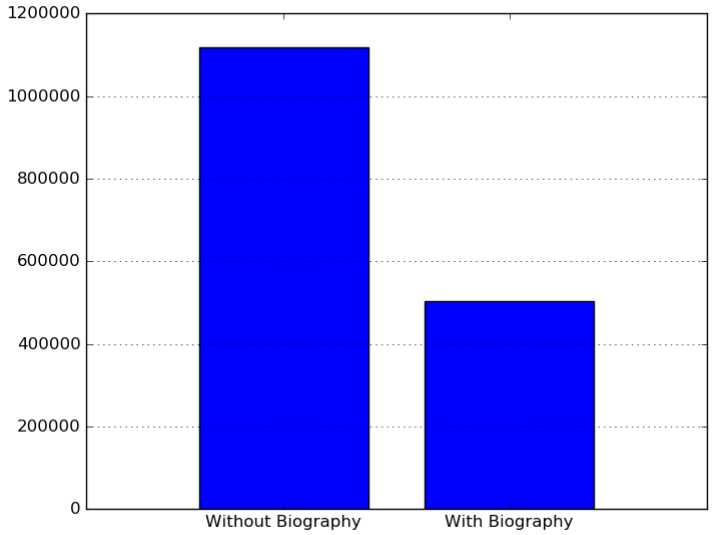

Figure 8.3: Distribution of StrepHit IEG Web Sources Corpus Items with Biographies and without Biographies

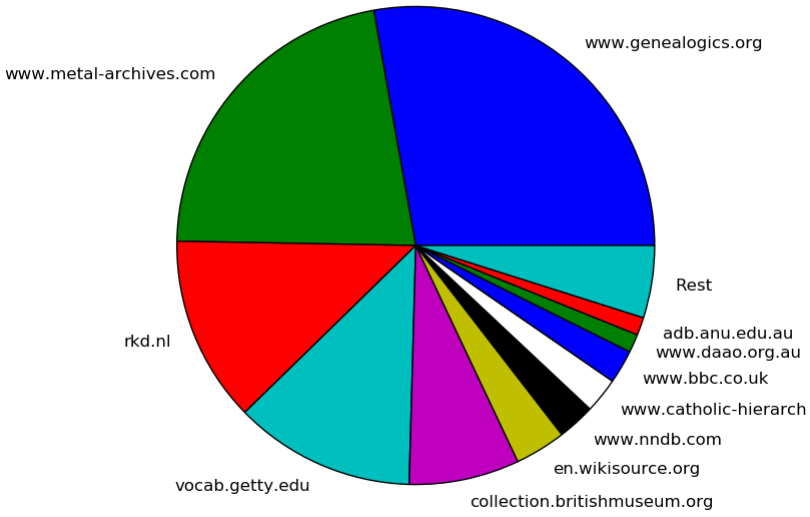

Figure 8.4: Pie Chart of StrepHit IEG Web Sources Corpus Items across Source Domains

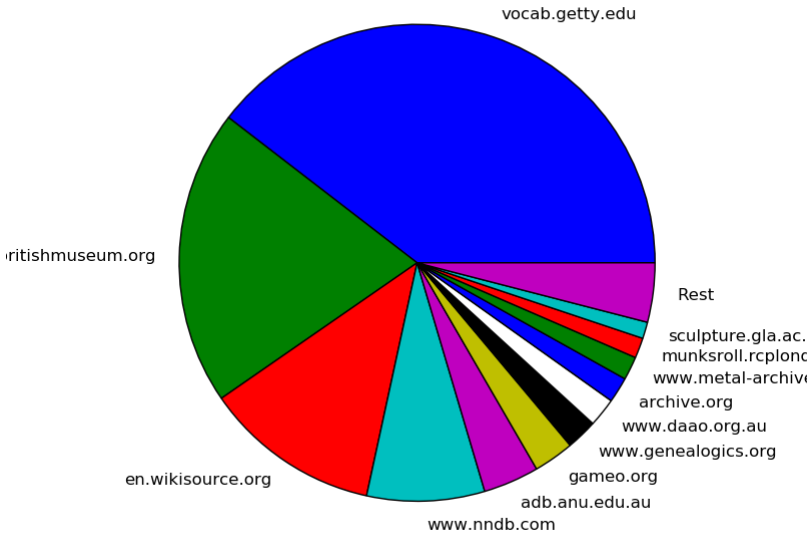

Figure 8.5: Pie Chart of StrepHit IEG Web Sources Corpus Biographies across Source Domains

8.6.4 Candidate Relations Set

The ranking is composed of verbs discovered via the corpus analysis module.[26] Each of them will trigger a set of Wikidata properties, depending on the number of FEs.

8.6.5 Semi-structured Development Dataset

During the corpus collection phase, we were asked to include sources with semi-structured data, typically names and dates. The result is a dataset that caters for the following Wikidata properties:

- birth name;[27]
- given name;[28]

[26] https://github.com/Wikidata/StrepHit/tree/master/strephit/corpus_analysis
[27] https://www.wikidata.org/wiki/Property:P1477
[28] https://www.wikidata.org/wiki/Property:P735

- family name;[29]
- pseudonym;[30]
- honorific suffix;[31]
- date of birth;[32]
- date of death;[33]
- sex or gender.[34]

Table 8.3 displays the amounts of references generated in the dataset, grouped by Web domains.

Domain	references
adb.anu.edu.au	6,262
collection.britishmuseum.org	1,7456
gameo.org	238
munksroll.rcplondon.ac.uk	418
archive.org	1,166
collection.cooperhewitt.org	366
sculpture.gla.ac.uk	247
dictionaryofarthistorians.org	103
en.wikisource.org	5,923
rkd.nl	2,416
structurae.net	254
viaf.org	387

[29] https://www.wikidata.org/wiki/Property:P734
[30] https://www.wikidata.org/wiki/Property:P742
[31] https://www.wikidata.org/wiki/Property:P1035
[32] https://www.wikidata.org/wiki/Property:P569
[33] https://www.wikidata.org/wiki/Property:P570
[34] https://www.wikidata.org/wiki/Property:P21

Domain	references
vocab.getty.edu	33,452
www.bbc.co.uk	9,847
www.museothyssen.org	240
www.newulsterbiography.co.uk	501
www.nndb.com	17,296
www.uni-stuttgart.de	2,465
www.wga.hu	1,577
yba.llgc.org.uk	39
Total	**100,266**

Table 8.3: Semi-structured development dataset references count across Web domains

8.7 Evaluation

Table 8.4 embeds the amount of references generated by StrepHit on its datasets and across Web sources. This gives a raw overview of the main goal of the project, namely to produce referenced Wikidata claims. Figure 8.6 displays the extraction outputs with respect to the confidence scores of linked entities: it is intended to highlight critical thresholds that should be used to achieve reasonable precision and recall trade-offs. We plot in Figure 8.7 standard performance values of the supervised classifier, computed on a random sample of lexical units. We observe different behaviors, depending on the lexical unit.

Domain	Confident	Supervised	Rule-based
adb.anu.edu.au	52,419	154,979	119,239
collection.britishmuseum.org	238,308	20,912	29,046
gameo.org	2,113	6,544	7,334

Domain	Confident	Supervised	Rule-based
munksroll.rcplondon.ac.uk	4,114	18,438	12,649
archive.org	8,103	39,062	30,146
collection.cooperhewitt.org	2,383	11,550	13,677
sculpture.gla.ac.uk	1,663	1,474	1,182
dictionaryofarthistorians.org	1,358	3,620	4,969
en.wikisource.org	51,232	227,346	209,411
rkd.nl	44,690	N.A.	N.A.
structurae.net	1,851	N.A.	N.A.
vocab.getty.edu	213,436	6,137	4,052
www.bbc.co.uk	54,070	2,109	2,254
www.brown.edu	N.A.	1,200	1,144
www.daao.org.au	N.A.	26,848	21,256
www.genealogics.org	19,870	10,186	14,536
www.metal-archives.com	N.A.	760	1,796
www.museothyssen.org	1,468	1,498	2,096
www.newulsterbiography.co.uk	3,284	3,438	5,379
www.nndb.com	106,782	26,402	30,101
www.uni-stuttgart.de	20,627	N.A.	N.A.
www.wga.hu	9,762	5,088	5,944
yba.llgc.org.uk	4,645	6,912	9,599
Total	**842,191**	**574,503**	**525,811**
Grand total		**1,942,505**	

Table 8.4: Statistics of referenced Wikidata claims across Web sources and StrepHit datasets

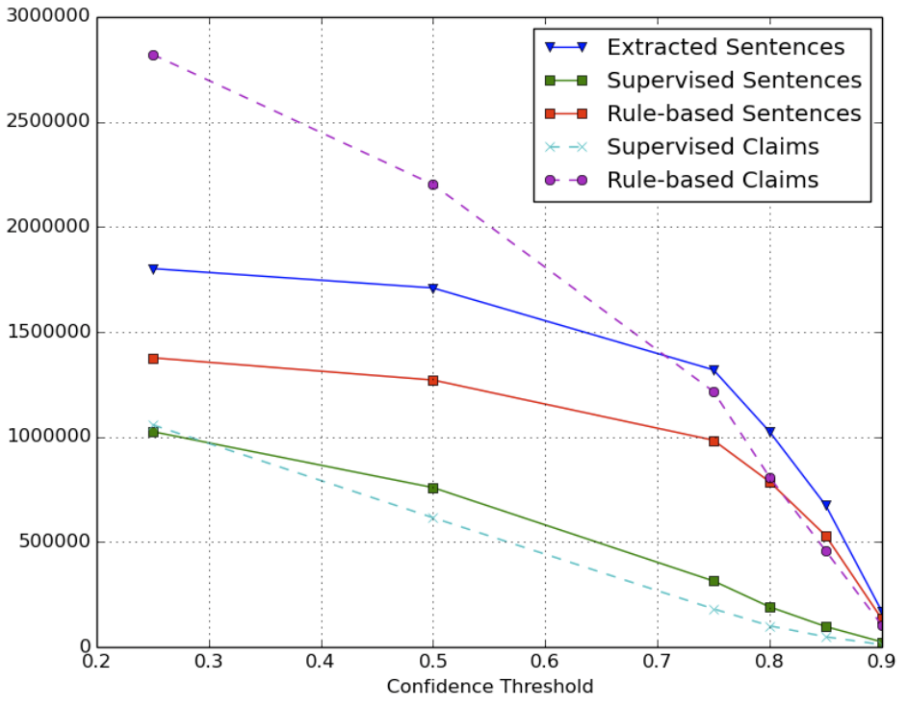

Figure 8.6: Amount of (1) sentences extracted from the input corpus, (2) classified sentences, and (3) generated Wikidata claims, with respect to confidence scores of linked entities

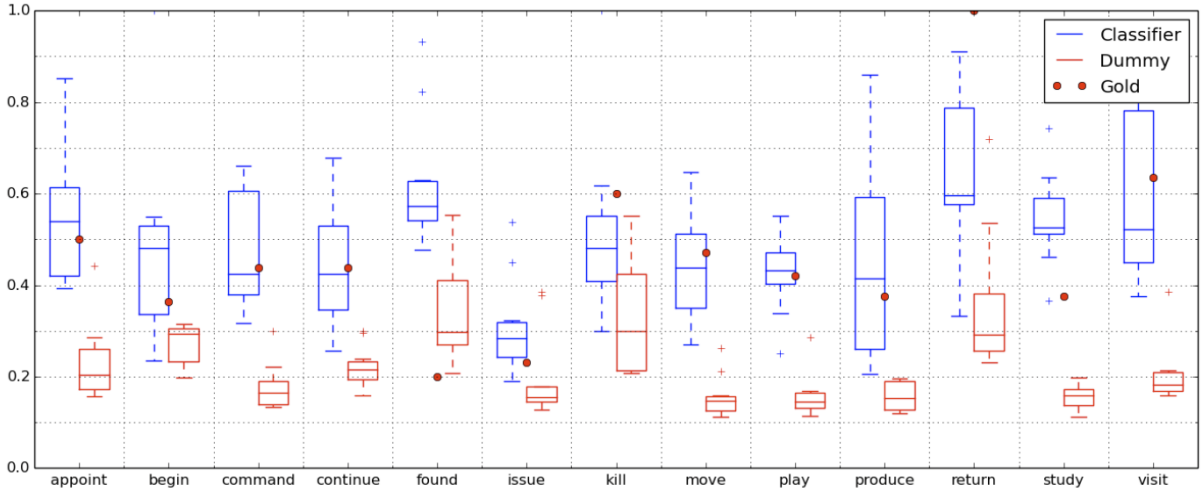

Figure 8.7: Performance values of the supervised classifier among a random sample of lexical units: (1) F1 scores via 10-fold cross validation, compared to a dummy classifier; (2) accuracy scores against a gold standard of 249 annotated sentences

8.7.1 Sample Statements

Machine-readable statements are expressed in the QuickStatements syntax.[35] The following list includes a random sample of correct statements that may serve as candidate for inclusion into Wikidata.

1. • P570 Q389547 +00000001837-01-01T00:00:00Z/9 S854 "http://www.bbc.co.uk/arts/yourpaintings/artists/hodges-charles-howard-17641837"
   • According to *BBC Your Paintings*, Charles Howard Hodges died in 1837

2. • Q17355708 P1477 "emma nicol" S854 "https://en.wikisource.org/wiki/Nicol,_Emma_(DNB00)"
   • According to the *Dictionary of National Biography*, Emma Nicol's birth name is "emma nicol"

3. • Q594729 P21 Q6581097 S854 "http://vocab.getty.edu/ulan/500110819"
   • According to the *Union List of Artist Names*, Anton Teichlein is a male

4. • Q215502 P742 "Morgan, Henry" S854 "http://collection.britishmuseum.org/id/person-institution/156902"
   • According to the *British Museum*, Henry Morgan's pseudonym is "Morgan, Henry"

5. • Q1562861 P569 +00000001939-08-21T00:00:00Z/11 S854 "http://www.nndb.com/people/103/000024031/"
   • According to the *Notable Names Database*, Clarence Williams III was born on August 21, 1939

[35] https://tools.wmflabs.org/wikidata-todo/quick_statements.php

6. • Q18526540 P569 +00000001815-02-24T00:00:00Z/11 S854 "http://adb.anu.edu.au/biography/barkly-sir-henry-2936"
   • According to the *Australian Dictionary of Biography*, Arthur Barkly was born on February 24, 1815

7. • Q16058737 P106 Q80687 S854 "https://ia902707.us.archive.org/1/items/biographicaldict08johnuoft/biographicaldict08johnuoftdjvu.txt"
   • According to *The Biographical Dictionary of America*, Charles Millard Pratt has been a secretary

8. • Q515632 P69 Q1068752 S854 "http://www.nndb.com/people/215/000042089/"
   • According to the *Notable Names Database*, Ossie Davis was educated at Howard University

9. • Q18922309 P937 Q777039 S854 "http://munksroll.rcplondon.ac.uk/Biography/Details/140"
   • According to the *Royal College of Physicians*, Henry Ashby has worked at Guy's Hospital

10. • Q4861627 P19 Q739700 S854 "http://www.bbc.co.uk/arts/yourpaintings/artists/barnett-freedman"
    • According to the *BBC Your Paintings* (now *Art UK*), Barnett Freedman was born in the East End of London

On the other hand, the following list shows a glimpse of wrong statements.

1. • Q3770981 P1477 "giusepe melani" S854 "http://vocab.getty.edu/ulan/500051662"
   • According to *Union List of Artist Names*, Giuseppe Melani's birth name is "giusepe melani"

- the source is wrong (possible typo)

2. - Q598060 P742 "Martyr Vermigli, Peter" S854 "http://collection.britishmuseum.org/id/person-institution/112005"
   - According to the *British Museum*, Peter Martyr Vermigli's pseudonym is "Martyr Vermigli, Peter"
   - debatable source assertion and Wikidata property label

3. - Q57297 P742 "E.W.L.T.; Ernesto Guglielmo Temple ; http://viaf.org/viaf/ S854 "http://www.uni-stuttgart.de/hi/gnt/dsi2/index.php?table_name=dsi&function=details&where_field=id&where_value=5752"
   - According to the *Database of Scientific Illustrators*, Wilhelm Tempel's pseudonym is "E.W.L.T.; Ernesto Guglielmo Temple ; http://viaf.org/viaf/45102696"
   - incorrect parsing of the source

4. - Q21454578 P463 Q42482 S854 "http://www.metal-archives.com/artists/Hugh_Gilmour/84280"
   - According to *Encyclopædia Metallum*, Hugh Gilmour was a member of the Iron Maiden
   - possibly homonymous subject (incorrect resolution), incorrect classification

5. - Q28144 P101 Q1193470 S854 "http://www.museothyssen.org/en/thyssen/ficha_artista/301"
   - According to the *Thyssen-Bornemisza Museum*, Willem Kalf's field of work is theme music
   - incorrect entity linking, incorrect classification

6. • Q3437676 P170 Q3908516 S854 "https://www.daao.org.au/bio/david-granger/"
   • According to *Design & Art Australia Online*, David Granger is the creator of entrepreneurship
   • homonymous subject (incorrect resolution), incorrect classification

8.7.2 Final Claim Correctness

We carried out an empirical evaluation over the final output results, by randomly sampling 48 claims from the supervised and the rule-based datasets. Since StrepHit is a pipeline with several components, we computed the accuracy of those responsible for the actual generation of claims. Results are presented in Table 8.5 and indicate the ratio of correct data for each of them, as well as the overall claim correctness.

Dataset	Claims	Linker	Classifier	Normalizer	Resolver	Overall
supervised	48	0.8125	0.781	1	0.285	0.638
rule-based	48	0.709	0.607	1	0.5	0.588

Table 8.5: Empirical claim correctness assessment

8.8 Resources

We provide below links to the project output.

- Codebase: https://github.com/Wikidata/StrepHit
- Documentation: https://www.mediawiki.org/wiki/StrepHit
- Web Sources Corpus

  - Development: http://it.dbpedia.org/downloads/strephit/web_sources_corpus/development_corpus.tar.gz
  - Production: http://it.dbpedia.org/downloads/strephit/web_sources_corpus/production_corpus.tar.gz

- Lexical database: http://it.dbpedia.org/downloads/strephit/lexical_db.json
- Web Sources Knowledge Base
  - Confident dataset: http://it.dbpedia.org/downloads/strephit/web_sources_knowledge_base/confident_dataset.qs.gz
  - Supervised dataset: http://it.dbpedia.org/downloads/strephit/web_sources_knowledge_base/supervised_dataset.qs.gz
  - Rule-based dataset: http://it.dbpedia.org/downloads/strephit/web_sources_knowledge_base/rule-based_dataset.qs.gz
- Unresolved entities
  - Confident: http://it.dbpedia.org/downloads/strephit/unresolved_entities/confident_unresolved.jsonl.gz
  - Supervised: http://it.dbpedia.org/downloads/strephit/unresolved_entities/supervised_unresolved.jsonl.gz
  - Rule-based: http://it.dbpedia.org/downloads/strephit/unresolved_entities/rule-based_unresolved.jsonl.gz
- Wiki Loves Monuments Italy prototype: http://it.dbpedia.org/downloads/strephit/wlm_italy_prototype/
- Italian Companies

  - Corpus: http://it.dbpedia.org/downloads/strephit/italian_companies_dataset/hackatoka_corpus.jsonl.gz
  - Lexical database: http://it.dbpedia.org/downloads/strephit/italian_companies_dataset/hackatoka_lexical_db.json
  - Dataset (not resolved to Wikidata): http://it.dbpedia.org/downloads/strephit/italian_companies_dataset/hackatoka_dataset.jsonl.gz

- All other resources at: http://it.dbpedia.org/downloads/strephit/

8.9 Challenges

Almost every challenge is technical, and most of them stem from NLP. We list them in order of decreasing impact. In general, scalability should be always taken into account during the software development.

Input Corpus. A relatively big input corpus from several sources introduces the need to cope with high language variability. Certain documents are written in old English, others stem from the OCR output of a paper scan, etc.

Target Lexical Database. It is unlikely that FrameNet would be a perfect fit for the data we aim at generating. This especially applies to the crowdsourcing part, since labels and definitions are minted by expert linguists, but cast to non-expert laymen. Hence, the major unplanned task (which affected the overall schedule of the project) was the construction of a suitable lexical database, since FrameNet failed to meet our needs. This had a negative impact in the most delicate planned task, namely building the crowdsourced training set.

Primary Sources Tool. Contributing to the maintenance of a third-party resource with generally low development activity can be time-consuming: it entails various tasks, from understanding possibly undocumented source code, to nudging the maintainers for addressing issues, all the way to accessing the machine that hosts the tool.

Crowdsourced Training Set. We had to sum extra issues related to the crowdsourcing platform and the nature of the input corpus. Respectively:

- high execution time for certain lexical units that are not trivial to annotate (at the time of writing this report, some jobs are still running);
- high percentage of sentence chunks that cannot be labeled with any frame element (more than 50% on average), which resulted in a relatively large amount of empty sentences even after the annotation.

This prevented us from reaching a sufficient amount of training samples, thus causing a generally low performance of the supervised classifier, depending on the lexical unit.

Dataset Serialization. Finding a general-purpose method to serialize the classification results into Wikidata assertions was impossible, since we needed to understand the intendend meaning of each Wikidata property, i.e., how it is used to represent the Wikidata world.

8.10 Side Projects

Besides StrepHit, we have been contributing to the following projects:

- primary sources tool, with 5 merged pull requests [36]

[36] https://github.com/Wikidata/primarysources/pull/86, https://github.com/

- Prototype import of Wiki Loves Monument Italy[37] into Wikidata[38]
- Sphinx[39] Python documentation builder[40]

Wikidata/primarysources/pull/87, https://github.com/Wikidata/primarysources/pull/97, https://github.com/Wikidata/primarysources/pull/100, https://github.com/Wikidata/primarysources/pull/102

[37] http://wikilovesmonuments.wikimedia.it/

[38] http://it.dbpedia.org/downloads/strephit/wlm_italy_prototype/, https://www.wikidata.org/wiki/Wikidata:Project_chat/Archive/2016/06#Importing_Wiki_Loves_Monuments_lists_into_Wikidata

[39] http://www.sphinx-doc.org/

[40] https://github.com/sphinx-doc/sphinx/pull/2444, https://github.com/Wikidata/StrepHit/tree/master/strephit/sphinx_wikisyntax

Bibliography

[1] Eneko Agirre, Enrique Alfonseca, Keith B. Hall, Jana Kravalova, Marius Pasca, and Aitor Soroa. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, May 31 - June 5, 2009, Boulder, Colorado, USA*, pages 19–27. The Association for Computational Linguistics, 2009.

[2] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 344–354. The Association for Computer Linguistics, 2015.

[3] Gabor Angeli, Julie Tibshirani, Jean Wu, and Christopher D. Manning. Combining distant and partial supervision for relation extraction. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha,*

Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 1556–1567. ACL, 2014.

[4] Alessio Palmero Aprosio, Claudio Giuliano, and Alberto Lavelli. Automatic expansion of dbpedia exploiting wikipedia cross-language information. In *The Semantic Web: Semantics and Big Data*, pages 397–411. Springer, 2013.

[5] Alessio Palmero Aprosio, Claudio Giuliano, and Alberto Lavelli. Towards an automatic creation of localized versions of dbpedia. In *The Semantic Web–ISWC 2013*, pages 494–509. Springer, 2013.

[6] Isabelle Augenstein, Diana Maynard, and Fabio Ciravegna. Relation extraction from the web using distant supervision. In Krzysztof Janowicz, Stefan Schlobach, Patrick Lambrix, and Eero Hyvönen, editors, *Knowledge Engineering and Knowledge Management - 19th International Conference, EKAW 2014, Linköping, Sweden, November 24-28, 2014. Proceedings*, volume 8876 of *Lecture Notes in Computer Science*, pages 26–41. Springer, 2014.

[7] Collin F. Baker. Framenet, current collaborations and future goals. *Language Resources and Evaluation*, 46(2):269–286, 2012.

[8] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The berkeley framenet project. In Christian Boitet and Pete Whitelock, editors, *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, COLING-ACL '98, August 10-14, 1998, Université de Montréal, Montréal, Quebec, Canada. Proceedings of the Conference.*, pages 86–90. Morgan Kaufmann Publishers / ACL, 1998.

[9] L. Douglas Baker and Andrew Kachites McCallum. Distributional clustering of words for text classification. In *Proceedings of the 21st*

Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 96–103. The Association for Computing Machinery, 1998.

[10] L. Bentivogli, P. Forner, C. Giuliano, A. Marchetti, E. Pianta, and K. Tymoshenko. Extending English ACE 2005 Corpus Annotation with Ground-truth Links to Wikipedia. In *23rd International Conference on Computational Linguistics*, pages 19–26, 2010.

[11] Jonathan Berant and Percy Liang. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 1415–1425. The Association for Computer Linguistics, 2014.

[12] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–165, 2009.

[13] Anders Björkelund, Love Hafdell, and Pierre Nugues. Multilingual semantic role labeling. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, pages 43–48. Association for Computational Linguistics, 2009.

[14] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In Jason Tsong-Li Wang, editor, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 1247–1250. ACM, 2008.

[15] Razvan C. Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In Diana McCarthy and Shuly Wintner, editors, *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy*. The Association for Computer Linguistics, 2006.

[16] Aljoscha Burchardt, Katrin Erk, Anette Frank, Andrea Kowalski, Sebastian Pado, and Manfred Pinkal. Salto–a versatile multi-level annotation tool. In *Proceedings of LREC 2006*, pages 517–520. Citeseer, 2006.

[17] Nicoletta Calzolari, Laura Pecchia, and Antonio Zampolli. Working on the italian machine dictionary: a semantic approach. In *Proceedings of the 5th Conference on Computational Linguistics*, volume 2, pages 49–52. Association for Computational Linguistics, 1973.

[18] I. Cantador, A. Bellogín, and P. Castells. A multilayer ontology-based hybrid recommendation model. *AI Communications*, 21(2):203–210, 2008.

[19] I. Cantador, A. Bellogín, and P. Castells. News@ hand: A semantic web approach to recommending news. In *Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 279–283. Springer, 2008.

[20] I. Cantador, A. Bellogín, and P. Castells. Ontology-based personalised and context-aware recommendations of news items. In *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on*, volume 1, pages 562–565. IEEE, 2008.

[21] I. Cantador, P. Castells, and A. Bellogín. An enhanced semantic layer for hybrid recommender systems: Application to news recom-

mendation. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 7(1):44–78, 2011.

[22] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. Toward an architecture for never-ending language learning. In Maria Fox and David Poole, editors, *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*. AAAI Press, 2010.

[23] Jon Chamberlain, Udo Kruschwitz, and Massimo Poesio. Constructing an anaphorically annotated corpus with non-experts: Assessing the quality of collaborative annotations. In *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, pages 57–62. Association for Computational Linguistics, 2009.

[24] Jon Chamberlain, Massimo Poesio, and Udo Kruschwitz. Phrase detectives: A web-based collaborative annotation game. *Proceedings of I-Semantics, Graz*, 2008.

[25] Nancy Chang, Praveen Paritosh, David Huynh, and Collin Baker. Scaling semantic frame annotation. In *Proceedings of The 9th Linguistic Annotation Workshop LAW IX, June 5, 2015, Denver, Colorado, USA*, pages 1–10. ACL, 2015.

[26] J. Chao, H. Wang, W. Zhou, W. Zhang, and Y. Yu. Tunesensor: A semantic-driven music recommendation service for digital photo albums. In *Proceedings of the 10th International Semantic Web Conference*, ISWC2011, October 2011.

[27] Jackie Chi Kit Cheung, Hoifung Poon, and Lucy Vanderwende. Probabilistic frame induction. *arXiv preprint arXiv:1302.4813*, 2013.

[28] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

[29] Bonaventura Coppola, Aldo Gangemi, Alfio Massimiliano Gliozzo, Davide Picca, and Valentina Presutti. Frame detection over the semantic web. In Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero Hyvönen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou, and Elena Paslaru Bontas Simperl, editors, *The Semantic Web: Research and Applications, 6th European Semantic Web Conference, ESWC 2009, Heraklion, Crete, Greece, May 31-June 4, 2009, Proceedings*, volume 5554 of *Lecture Notes in Computer Science*, pages 126–142. Springer, 2009.

[30] Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In Jason Eisner, editor, *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*, pages 708–716. ACL, 2007.

[31] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In Marta Sabou, Eva Blomqvist, Tommaso Di Noia, Harald Sack, and Tassilo Pellegrini, editors, *I-SEMANTICS 2013 - 9th International Conference on Semantic Systems, ISEM '13, Graz, Austria, September 4-6, 2013*, pages 121–124. ACM, 2013.

[32] Dipanjan Das, Desai Chen, André F. T. Martins, Nathan Schneider, and Noah A. Smith. Frame-semantic parsing. *Computational Linguistics*, 40(1):9–56, 2014.

[33] Dipanjan Das, André FT Martins, and Noah A Smith. An exact dual decomposition algorithm for shallow semantic parsing with constraints. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 209–217. Association for Computational Linguistics, 2012.

[34] Gerard de Melo and Gerhard Weikum. Menta: Inducing multilingual taxonomies from wikipedia. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 1099–1108. ACM, 2010.

[35] Filipe de Sá Mesquita, Jordan Schmidek, and Denilson Barbosa. Effectiveness and efficiency of open relation extraction. 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 447–457. ACL, 2013.

[36] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990.

[37] T. Di Noia, R. Mirizzi, V.C. Ostuni, D. Romito, and M. Zanker. Linked open data to support content-based recommender systems. In *Proceedings of the 8th International Conference on Semantic Systems*, pages 1–8. ACM, 2012.

[38] Li Ding, Timothy Lebo, John S Erickson, Dominic DiFranzo, Gregory Todd Williams, Xian Li, James Michaelis, Alvaro Graves,

Jin Guang Zheng, Zhenning Shangguan, et al. Twc logd: A portal for linked open government data ecosystems. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(3):325–333, 2011.

[39] 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 Sofus A. Macskassy, Claudia Perlich, Jure Leskovec, Wei Wang, and Rayid Ghani, editors, *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. ACM, 2014.

[40] Arnab Dutta, Christian Meilicke, and Heiner Stuckenschmidt. Enriching structured knowledge with open information. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, pages 267–277, 2015.

[41] Maud Ehrmann, Francesco Cecconi, Daniele Vannella, John McCrae, Philipp Cimiano, and Roberto Navigli. Representing multilingual data as linked data: the case of babelnet 2.0. In *Proceedings of the 9th Language Resources and Evaluation Conference*, 2014.

[42] Fredo Erxleben, Michael Günther, Markus Krötzsch, Julian Mendez, and Denny Vrandecic. Introducing wikidata to the linked data web. In *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*, pages 50–65, 2014.

[43] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. 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 1535–1545. ACL, 2011.

[44] Manaal Faruqui and Shankar Kumar. Multilingual open relation extraction using cross-lingual projection. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1351–1356, 2015.

[45] Christiane Fellbaum. *Wordnet: an Electronic Lexical Database*. MIT Press Cambridge, 1998.

[46] Charles Fillmore. Frame semantics. *Linguistics in the morning calm*, pages 111–137, 1982.

[47] Charles J. Fillmore. Frame Semantics and the nature of language. In *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language*, pages 20–32. Blackwell Publishing, 1976.

[48] Charles J. Fillmore, Collin F. Baker, and Hiroaki Sato. The FrameNet Database and Software Tools. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)*, pages 1157–1160, Las Palmas, Spain, 2002.

[49] Tiziano Flati, Daniele Vannella, Tommaso Pasini, and Roberto Navigli. Two is bigger (and better) than one: the wikipedia bitaxonomy project. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014.

[50] Marco Fossati, Claudio Giuliano, and Sara Tonelli. Outsourcing framenet to the crowd. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 2: Short Papers*, pages 742–747. The Association for Computer Linguistics, 2013.

[51] Aldo Gangemi, Andrea Giovanni Nuzzolese, Valentina Presutti, Francesco Draicchio, Alberto Musetti, and Paolo Ciancarini. Automatic typing of dbpedia entities. In *The Semantic Web–ISWC 2012*, pages 65–81. Springer, 2012.

[52] Aldo Gangemi and Valentina Presutti. Towards a pattern science for the semantic web. *Semantic Web*, 1(1-2):61–68, 2010.

[53] Aldo Gangemi, Valentina Presutti, Diego Reforgiato Recupero, Andrea Giovanni Nuzzolese, Francesco Draicchio, and Misael Mongiovìb. Semantic web machine reading with fred. *Semantic Web*, 2016. Under Review.

[54] Claudio Giuliano, Alfio Massimiliano Gliozzo, and Carlo Strapparava. Kernel methods for minimally supervised WSD. *Computational Linguistics*, 35(4):513–528, 2009.

[55] Jonathan Gray, Lucy Chambers, and Liliana Bounegru. *The Data Journalism Handbook*. O'Reilly Media, Inc., 2012.

[56] Marco Guerini, Carlo Strapparava, and Oliviero Stock. Ecological evaluation of persuasive messages using google adwords. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, volume abs/1204.5369 of *ACL2012*, July 2012.

[57] Bernhard Haslhofer, Elaheh Momeni, Manuel Gay, and Rainer Simon. Augmenting europeana content with linked data resources. In *Pro-*

ceedings of the 6th International Conference on Semantic Systems, page 40. ACM, 2010.

[58] Conor Hayes, Pádraig Cunningham, and Paolo Massa. An on-line evaluation framework for recommender systems. Technical Report TCD-CS-2002-19, Trinity College Dublin, Department of Computer Science, 2002.

[59] Michael Heilman and Noah A. Smith. Extracting Simplified Statements for Factual Question Generation. In *Proceedings of QG2010: The Third Workshop on Question Generation*, Pittsburgh, PA, USA, 2010.

[60] Sebastian Hellmann, Jens Lehmann, Sören Auer, and Martin Brümmer. Integrating NLP using linked data. In *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part II*, pages 98–113, 2013.

[61] Daniel Hernández, Aidan Hogan, and Markus Krötzsch. Reifying RDF: what works well with wikidata? In *Proceedings of the 11th International Workshop on Scalable Semantic Web Knowledge Base Systems co-located with 14th International Semantic Web Conference (ISWC 2015), Bethlehem, PA, USA, October 11, 2015.*, pages 32–47, 2015.

[62] Rinke Hoekstra. *Ontology Representation - Design Patterns and Ontologies that Make Sense*, volume 197 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009.

[63] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. YAGO2: A spatially and temporally enhanced knowledge base from wikipedia. *Artif. Intell.*, 194:28–61, 2013.

[64] Jisup Hong and Collin F Baker. How Good is the Crowd at "real" WSD? In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 30–37, 2011.

[65] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich. *Recommender Systems: An Introduction*. Cambridge University Press, 2011.

[66] Richard Johansson and Pierre Nugues. Lth: Semantic structure extraction using nonprojective dependency trees. In *Proceedings of the 4th international workshop on semantic evaluations*, pages 227–230. The Association for Computational Linguistics, 2007.

[67] Richard Johansson and Pierre Nugues. Dependency-based semantic role labeling of propbank. 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 69–78. ACL, 2008.

[68] Nicolas Kaufmann, Thimo Schulze, and Daniel Veit. More than fun and money. worker motivation in crowdsourcing–a study on mechanical turk. In *Proceedings of the Seventeenth Americas Conference on Information Systems, Detroit, MI*, 2011.

[69] Shailesh Kochhar, Stefano Mazzocchi, and Praveen Paritosh. The anatomy of a large-scale human computation engine. In *Proceedings of the acm sigkdd workshop on human computation*, pages 10–17. ACM, 2010.

[70] Dimitris Kontokostas, Charalampos Bratsas, Sören Auer, Sebastian Hellmann, Ioannis Antoniou, and George Metakides. Internationalization of linked data: The case of the greek dbpedia edition. *J. Web Sem.*, 15:51–61, 2012.

[71] Meghana Kshirsagar, Sam Thomson, Nathan Schneider, Jaime G. Carbonell, Noah A. Smith, and Chris Dyer. Frame-semantic role labeling with heterogeneous annotations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, pages 218–224. The Association for Computational Linguistics, 2015.

[72] Ivo Lašek. Dc proposal: Model for news filtering with named entities. In Lora Aroyo, Chris Welty, Harith Alani, Jamie Taylor, Abraham Bernstein, Lalana Kagal, Natasha Noy, and Eva Blomqvist, editors, *The Semantic Web – ISWC 2011*, volume 7032 of *Lecture Notes in Computer Science*, pages 309–316. Springer Berlin / Heidelberg, 2011.

[73] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6(2):167–195, 2015.

[74] L. Li, W. Chu, J. Langford, and R.E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

[75] P. Lops, M. Gemmis, and G. Semeraro. Content-based recommender systems: State of the art and trends. *Recommender Systems Handbook*, pages 73–105, 2011.

[76] Yuanhua Lv, Taesup Moon, Pranam Kolari, Zhaohui Zheng, Xuanhui Wang, and Yi Chang. Learning to model relatedness for news rec-

ommendation. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 57–66, New York, NY, USA, 2011. ACM.

[77] Lluís Màrquez, Xavier Carreras, Kenneth C. Litkowski, and Suzanne Stevenson. Semantic role labeling: An introduction to the special issue. *Computational Linguistics*, 34(2):145–159, 2008.

[78] Winter Mason and Siddharth Suri. Conducting behavioral research on amazon's mechanical turk. *Behavior Research Methods*, 44:1–23, 2012.

[79] Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. Open language learning for information extraction. In Jun'ichi Tsujii, James Henderson, and Marius Pasca, editors, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 523–534. ACL, 2012.

[80] S.M. McNee, J. Riedl, and J.A. Konstan. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI'06 extended abstracts on Human factors in computing systems*, pages 1097–1101. ACM, 2006.

[81] Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: shedding light on the web of documents. In Chiara Ghidini, Axel-Cyrille Ngonga Ngomo, Stefanie N. Lindstaedt, and Tassilo Pellegrini, editors, *Proceedings the 7th International Conference on Semantic Systems, I-SEMANTICS 2011, Graz, Austria, September 7-9, 2011*, ACM International Conference Proceeding Series, pages 1–8. ACM, 2011.

[82] Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, I-Semantics '11, pages 1–8, New York, NY, USA, 2011. ACM.

[83] David Milne and Ian H. Witten. Learning to link with Wikipedia. In *CIKM '08: Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518, NY, USA, 2008. ACM.

[84] Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity linking meets word sense disambiguation: a unified approach. *TACL*, 2:231–244, 2014.

[85] Vivi Nastase and Michael Strube. Transforming wikipedia into a large scale multilingual concept network. *Artificial Intelligence*, 194:62–85, 2013.

[86] Vivi Nastase, Michael Strube, Benjamin Boerschinger, Cäcilia Zirn, and Anas Elghafari. Wikinet: A very large scale multi-lingual concept network. In *Proceedings of the 5th Language Resources and Evaluation Conference*, 2010.

[87] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: the automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *AI*, 193:217–250, 2012.

[88] Matteo Negri, Luisa Bentivogli, Yashar Mehdad, Danilo Giampiccolo, and Alessandro Marchetti. Divide and conquer: crowdsourcing the creation of cross-lingual textual entailment corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 670–679, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[89] Vinh Nguyen, Olivier Bodenreider, and Amit Sheth. Don't like rdf reification?: making statements about statements using singleton property. In *Proceedings of the 23rd international conference on World wide web*, pages 759–770. ACM, 2014.

[90] Andrea Giovanni Nuzzolese, Aldo Gangemi, and Valentina Presutti. Gathering lexical linked data and knowledge patterns from framenet. In Mark A. Musen and Óscar Corcho, editors, *Proceedings of the 6th International Conference on Knowledge Capture (K-CAP 2011), June 26-29, 2011, Banff, Alberta, Canada*, pages 41–48. ACM, 2011.

[91] Andrea Giovanni Nuzzolese, Aldo Gangemi, Valentina Presutti, and Paolo Ciancarini. Fine-tuning triplification with semion. In *EKAW workshop on Knowledge Injection into and Extraction from Linked Data (KIELD2010)*, pages 2–14, 2010.

[92] Sebastian Padó and Mirella Lapata. Cross-lingual annotation projection for semantic roles. *Journal of Artificial Intelligence Research*, 36(1):307–340, 2009.

[93] Martha Palmer, Dan Gildea, and Paul Kingsbury. The Proposition Bank: A Corpus Annotated with Semantic Roles. *Computational Linguistics*, 31(1), 2005.

[94] Heiko Paulheim and Christian Bizer. Type inference on noisy rdf data. In *The Semantic Web–ISWC 2013*, pages 510–525. Springer, 2013.

[95] Michael Pazzani and Daniel Billsus. Content-based recommendation systems. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 325–341. Springer Berlin / Heidelberg, 2007.

[96] Fernando C. N. Pereira, Naftali Tishby, and Lillian Lee. Distributional clustering of english words. *CoRR*, abs/cmp-lg/9408011, 1994.

[97] Silvio Peroni, Aldo Gangemi, and Fabio Vitali. Dealing with markup semantics. In *Proceedings the 7th International Conference on Semantic Systems, I-SEMANTICS 2011, Graz, Austria, September 7-9, 2011*, pages 111–118, 2011.

[98] Emanuele Pianta, Christian Girardi, and Roberto Zanoli. The textpro tool suite. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco*, 2008.

[99] Aleksander Pohl. Classifying the wikipedia articles into the opencyc taxonomy. In *Proceedings of the Web of Linked Entities Workshop in conjuction with the 11th International Semantic Web Conference*, 2012.

[100] Carl Pollard and Ivan A. sag. *Information-based Syntax and Semantics: vol. 1: Fundamentals*. Center for the Study of Language and Information, Stanford, CA, USA, 1988.

[101] Carl Pollard and Ivan A sag. *Head-driven Phrase Structure Grammar*. University of Chicago Press, 1994.

[102] Simone Paolo Ponzetto and Michael Strube. Deriving a large scale taxonomy from wikipedia. In *Proceeding of AAAI*, volume 7, pages 1440–1445, 2007.

[103] Simone Paolo Ponzetto and Michael Strube. Taxonomy induction based on a collaboratively built knowledge repository. *Artificial Intelligence*, 175(9-10):1737–1756, 2011.

[104] Valentina Presutti, Sergio Consoli, Andrea Giovanni Nuzzolese, Diego Reforgiato Recupero, Aldo Gangemi, Ines Bannour, and Haïfa Zargayouna. Uncovering the semantics of wikipedia pagelinks. In Krzysztof Janowicz, Stefan Schlobach, Patrick Lambrix, and Eero Hyvönen, editors, *Knowledge Engineering and Knowledge Management - 19th International Conference, EKAW 2014, Linköping, Sweden, November 24-28, 2014. Proceedings*, volume 8876 of *Lecture Notes in Computer Science*, pages 413–428. Springer, 2014.

[105] Valentina Presutti, Francesco Draicchio, and Aldo Gangemi. Knowledge extraction based on discourse representation theory and linguistic frames. In *Knowledge Engineering and Knowledge Management*, pages 114–129. Springer, 2012.

[106] Valentina Presutti, Andrea Giovanni Nuzzolese, Sergio Consoli, Diego Reforgiato Recupero, and Aldo Gangemi. From hyperlinks to semantic web properties using open knowledge extraction. *Semantic Web*, Preprint(Preprint):1–28, 2016.

[107] Vasin Punyakanok, Dan Roth, and Wen-tau Yih. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2):257–287, 2008.

[108] Jacobo Rouces, Gerard de Melo, and Katja Hose. Framebase: Representing n-ary relations using semantic frames. In Fabien Gandon, Marta Sabou, Harald Sack, Claudia d'Amato, Philippe Cudré-Mauroux, and Antoine Zimmermann, editors, *The Semantic Web. Latest Advances and New Domains - 12th European Semantic Web Conference, ESWC 2015, Portoroz, Slovenia, May 31 - June 4, 2015. Proceedings*, volume 9088 of *Lecture Notes in Computer Science*, pages 505–521. Springer, 2015.

[109] Jacobo Rouces, Gerard de Melo, and Katja Hose. Integrating heterogeneous knowledge with framebase. *Semantic Web*, 2016. Under Review.

[110] J. Ruppenhofer, C. Sporleder, R. Morante, C. Baker, and M. Palmer. Semeval-2010 task 10: Linking events and their participants in discourse. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 45–50. Association for Computational Linguistics, 2010.

[111] Josef Ruppenhofer, Michael Ellsworth, Miriam R.L. Petruck, Christopher R. Johnson, and Jan Scheffczyk. *FrameNet II: Extended Theory and Practice.* Available at http://framenet.icsi.berkeley.edu/book/book.html, 2006.

[112] Thomas Schmidt. The kicktionary revisited. In Angelika Storrer, Alexander Geyken, Alexander Siebert, and Kay-Michael Würzner, editors, *Text Resources and Lexical Knowledge. Selected Papers from the 9th Conference on Natural Language Processing, KONVENS 2008, Berlin, Germany*, pages 239–251. Mouton de Gruyter, 2008.

[113] D. Shahaf and C. Guestrin. Connecting the dots between news articles. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 623–632. ACM, 2010.

[114] P. Shoval, V. Maidel, and B. Shapira. An ontology-content-based filtering method. *International Journal on Information Theories and Applications*, 15(4):303–314, 2008.

[115] Daniel Dominic Sleator and David Temperley. Parsing english with a link grammar. *CoRR*, abs/cmp-lg/9508004, 1995.

[116] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 254–263. Association for Computational Linguistics, 2008.

[117] Fabian M Suchanek, Georgiana Ifrim, and Gerhard Weikum. Leila: Learning to extract information by linguistic analysis. In *Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pages 18–25, 2006.

[118] 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*, pages 697–706. ACM, 2007.

[119] Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. Multi-instance multi-label learning for relation extraction. In Jun'ichi Tsujii, James Henderson, and Marius Pasca, editors, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 455–465. ACL, 2012.

[120] Thomas Pellissier Tanon, Denny Vrandecic, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher. From freebase to wikidata: The great migration. In Jacqueline Bourdeau, Jim Hendler, Roger Nkambou, Ian Horrocks, and Ben Y. Zhao, editors, *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 1419–1428. ACM, 2016.

[121] A. Thalhammer, T. Ermilov, K. Nyberg, A. Santoso, and J. Domingue. Moviegoer - semantic social recommendations and personalized

location-based offers. In *Proceedings of the 10th International Semantic Web Conference*, ISWC2011, October 2011.

[122] E.G. Toms. Serendipitous information retrieval. In *Proceedings of the First DELOS Network of Excellence Workshop on Information Seeking, Searching and Querying in Digital Libraries*, pages 11–12, 2000.

[123] Sara Tonelli, Claudio Giuliano, and Kateryna Tymoshenko. Wikipedia-based WSD for multilingual frame annotation. *Artificial Intelligence*, 194:203–221, 2013.

[124] Luis Von Ahn, Mihir Kedia, and Manuel Blum. Verbosity: a game for collecting common-sense facts. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 75–78. ACM, 2006.

[125] Denny Vrandecic and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, 2014.

[126] Fei Wu and Daniel S. Weld. Open information extraction using wikipedia. In Jan Hajic, Sandra Carberry, and Stephen Clark, editors, *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 118–127. The Association for Computer Linguistics, 2010.

[127] Xuchen Yao and Benjamin Van Durme. Information extraction over structured data: Question answering with freebase. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 956–966. The Association for Computer Linguistics, 2014.

[128] Shenghuo Zhu, Kai Yu, Yun Chi, and Yihong Gong. Combining content and link for classification using matrix factorization. In Wessel Kraaij, Arjen P. de Vries, Charles L. A. Clarke, Norbert Fuhr, and Noriko Kando, editors, *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007*, pages 487–494. ACM, 2007.

[129] Cai-Nicolas Ziegler, Georg Lausen, and Lars Schmidt-Thieme. Taxonomy-driven computation of product recommendations. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 406–415, New York, NY, USA, 2004. ACM.

[130] Cäcilia Zirn, Vivi Nastase, and Michael Strube. Distinguishing between instances and classes in the wikipedia taxonomy. In *Proceedings of the 5th European Semantic Web Conference*, pages 376–387, 2008.