



D2.3.2

Multilingual resources and evaluation of knowledge modelling - v2

Genevieve Gorrell, Johann Petrak, Kalina Bontcheva – USFD
Guy Emerson, Thierry Declerck – DFKI

Abstract.

FP7-ICT Strategic Targeted Research Project (STREP) ICT-2011-287863 TrendMiner
Deliverable D2.3.2 (WP2)

This deliverable presents first results of the evaluation of the final version of the LODIE system. The system has been extended to include a significant array of state of the art and novel technology, and has been renamed YODIE to distinguish it from the earlier version. New technological contributions are outlined. German, Bulgarian and Hindi are delivered.

In a second part, the deliverable presents a Bayesian probabilistic model, which can simultaneously combine polarity scores from several data sources and estimate the quality of each source.

In its last part, the deliverable points to sections of TrendMiner deliverables in which the new partners of the TrendMiner consortium describe evaluation results of their work.

Keyword list: information extraction, natural language processing, ontologies, polarity lexicons, opinion mining

Project	TrendMiner No. 287863
Delivery Date	October 31, 2014
Contractual Date	October 31, 2014
Nature	Prototype
Reviewed By	N/A
Web links	http://demos.gate.ac.uk/trendminer/rest/service/annotate
Dissemination	PU

TrendMiner Consortium

This document is part of the TrendMiner research project (No. 287863), partially funded by the FP7-ICT Programme.

DFKI GmbH

Language Technology Lab
Stuhlsatzenhausweg 3
D-66123 Saarbrücken
Germany
Contact person: Thierry Declerck
E-mail: declerck@dfki.de

University of Southampton

Southampton SO17 1BJ
UK
Contact person: Mahesan Niranjan
E-mail: mn@ecs.soton.ac.uk

Internet Memory Research

45 ter rue de la Révolution
F-93100 Montreuil
France
Contact person: France Lafarges
E-mail: contact@internetmemory.org

Eurokleis S.R.L.

Via Giorgio Baglivi, 3
Roma RM
00161 Italy
Contact person: Francesco Bellini
E-mail: info@eurokleis.com

Institute of Computer Science Polish Academy of Sciences

5 Jana Kazimierza Str
01-248 Warsaw
Poland
Contact person: Maciej Ogrodniczuk
E-mail: Maciej.Ogrodniczuk@ipipan.waw.pl

Universidad Carlos III de Madrid

Av. Universidad, 30
28911 Madrid
Spain
Contact person: Paloma Martínez Fernández
E-mail: pmf@inf.uc3m.es

University of Sheffield

Department of Computer Science
Regent Court, 211 Portobello St.
Sheffield S1 4DP
UK
Contact person: Kalina Bontcheva
E-mail: K.Bontcheva@dcs.shef.ac.uk

Ontotext AD

Polygraphia Office Center fl.4,
47A Tsarigradsko Shosse,
Sofia 1504, Bulgaria
Contact person: Atanas Kiryakov
E-mail: naso@sirma.bg

Sora Ogris and Hofinger GmbH

Bennogasse 8/2/16
A-1080 Wien
Austria
Contact person: Christoph Hofinger
E-mail: ch@sora.at

Hardik Fintrade Pvt Ltd.

227, Shree Ram Cloth Market,
Opposite Manilal Mansion,
Revdi Bazar, Ahmedabad 380002
India
Contact person: Suresh Aswani
E-mail: m.aswani@hardikgroup.com

DAEDALUS - DATA, DECISIONS AND LANGUAGE, S. A.

C/ López de Hoyos 15, 3
28006 Madrid
Spain
Contact person: José Luis Martínez Fernández
E-mail: jmartinez@daedalus.es

Research Institute for Linguistics of the Hungarian Academy of Sciences

Benczr u. 33,
1068 Budapest Hungary
Contact person: Tamás Váradi
E-mail: varadi.tamas@nytud.mta.hu

Chapter 1

Introduction

This deliverable presents in the first chapter results of the evaluation of the final version of the LODIE system. The system has been extended to include a significant array of state of the art and novel technology, and has been renamed YODIE to distinguish it from the earlier version. New technological contributions are outlined. German, Bulgarian and Hindi are delivered. Evaluation is performed in contrast with other competitive current systems for named entity linking. Results are provided, with discussion of the ways in which our success was achieved, especially in the use-case context of tweets, where our technological contribution is particularly relevant.

The second chapter presents work on heuristics developed to estimate the quality of polarity lexicons, since many approaches to sentiment analysis rely on a lexicon that labels words with a prior polarity. This is particularly true for languages other than English, where labelled training data is not easily available. Existing efforts to produce such lexicons exist, and to avoid duplicated effort, a principled way to combine multiple resources is required. In this chapter, we introduce a Bayesian probabilistic model, which can simultaneously combine polarity scores from several data sources and estimate the quality of each source. We apply this algorithm to a set of four German sentiment lexicons, to produce the SentiMerge lexicon, which we make publically available. In a simple classification task, we show that this lexicon outperforms each of the underlying resources, as well as a majority vote model. We expect our approach to work similarly when applied to other languages. Parts of the text used in this deliverable have been used for a successful paper submission at the Coling 2014 LGLP Workshop¹

In the third chapter, we present very briefly work done by the four new partners of the project (UC3M, Daedalus, RILMTA and IIPAN), and points to evaluation results they have been described in the Deliverable 10.1: Newly generated domainspecific language data and tools.

¹<http://lg-lp.info/>.

Chapter 2

Information Extraction

2.1 Introduction

Information Extraction (IE) [Car97, App99] is a form of natural language analysis, which takes textual content as input and extracts fixed-type, unambiguous snippets as output. The extracted data may be used directly for display to users (e.g. a list of named entities mentioned in a document), for storing in a database for later analysis, or for improving information search and other information access tasks.

Named Entity Recognition (NER) is one of the key information extraction tasks, which is concerned with identifying names of entities such as people, locations, organisations and products. It is typically broken down into two main phases: *entity detection* and *entity typing* (also called classification) [GS96]. A follow-up step to NER is Named Entity Linking (NEL), which links entity mentions within the same document (also known as entity disambiguation) [HC97], or in other resources (also known as entity resolution) [RMD13]. Typically, state-of-the-art NER and NEL systems are developed and evaluated on news articles and other carefully written, longer content [RR09, RMD13].

In recent years, social media – and microblogging in particular – have established themselves as high-value, high-volume content, which organisations increasingly wish to analyse automatically. Currently, the leading microblogging platform is Twitter [JSFT07], which has around 288 million active users, posting over 500 million tweets a day,¹ and has the fastest growing network in terms of active usage.²

Reliable entity recognition and linking of user-generated content is an enabler for other information extraction tasks (e.g. relation extraction), as well as opinion mining [MBR12], and summarisation [RBH13]. It is relevant in many application contexts [DYJ13], including knowledge management, competitor intelligence, customer relation management, eBusiness, eScience, eHealth, and eGovernment.

Information extraction over microblogs has only recently become an active research

¹http://news.cnet.com/8301-1023_3-57541566-93/report-twitter-hits-half-a-billion-tweets-a-day

²<http://globalwebindex.net/thinking/social-platforms-gwi-8-update-decline-of-local-social-media-platforms>

topic [RSD⁺13], following early experiments which showed this genre to be extremely challenging for state-of-the-art algorithms [DMAB13a]. For instance, named entity recognition methods typically have 85-90% accuracy on longer texts, but 30-50% on tweets [RCME11, LZW⁺12]. First, the shortness of microblogs (maximum 140 characters for tweets) makes them hard to interpret. Consequently, ambiguity is a major problem since semantic annotation methods cannot easily make use of coreference information. Unlike longer news articles, there is a low amount of discourse information per microblog document, and threaded structure is fragmented across multiple documents, flowing in multiple directions. Second, microtexts exhibit much more language variation, tend to be less grammatical than longer posts, contain unorthodox capitalisation, and make frequent use of emoticons, abbreviations and hashtags, which can form an important part of the meaning. To combat these problems, research has focused on microblog-specific information extraction algorithms (e.g. named entity recognition for Twitter using CRFs [RCME11] or hybrid methods [vERT13]). Particular attention is given to microtext normalisation [HB11], as a way of removing some of the linguistic noise prior to part-of-speech tagging and entity recognition.

In this deliverable, we present our evaluation of YODIE (previously LODIE), the combined named entity detection and linking system developed as part of WP2 in TrendMiner [AGBP13]. YODIE not only delivers a solid performance across a variety of different content types, but also aims to cope with the particular issues raised by the microblog genre. We build on earlier deliverables [AGBP13, AGB⁺12] describing the YODIE architecture, outlining the additional disambiguation features added since [AGBP13], particularly highlighting the ways in which YODIE has been tailored to noisy social media content.

In the next chapter, we present the competing systems against which we compare YODIE's performance. Following that, we describe the corpora and metrics we have chosen to demonstrate performance, and our rationale for choosing them. After presenting our evaluation, we conclude with a discussion of the contributing factors in YODIE's success.

2.1.1 Relevance to TrendMiner

The majority of the work reported in this chapter is describing the results of Task 2.4 Evaluation and builds on the results of Tasks 2.1 (Multilingual knowledge and lexical acquisition for customizing Ontology Schema) and 2.3 (Multilingual, ontology-based IE from stream media: entities, events, sentiment, and trends). Since the focus is on evaluation, this report describes the datasets, systems, experiments, and results carried out as part of Task 2.4.

Relevance to project objectives

The work reported in this chapter provides the evaluation of the TrendMiner methods for information extraction and disambiguation over streaming media.

Relation to other workpackages

Information extraction and disambiguation are requirements of the two use cases (WP6 and WP7) as well as being preparatory steps required for the summarization work carried out in WP4. The results are visualised in the integrated platform from WP5.

Chapter 3

Systems Compared To

3.1 Introduction

There are a number of existing methods and services for entity linking and disambiguation, against which it is informative to compare YODIE. An attempt has been made to comprehensively cover the available NEL systems, insofar as it is feasible. Not all prominent systems have been included; OpenCalais, for example, does not universally provide links to DBpedia, making it infeasible to compare with the other systems. DBpedia (<http://dbpedia.org>) is a community effort to produce a semantic representation for some of the information contained in Wikipedia. Most of the DBpedia data is automatically derived, so it is not always very reliable. However, it remains the most widely used shared knowledge base and the best option in terms of allowing a comprehensive comparison.

Another system that we excluded is AlchemyAPI, which has terms of service preventing researchers from using it for comparative evaluation and publishing results that include Alchemy API.

In this chapter, we describe the systems, including our own. LODIE has been described in D2.2.2 [AGBP13], so will not be described here other than briefly to position it with respect to the other systems, and outline ways in which its new version, YODIE, differs in terms of features and methods used. Then, we describe each of the comparison systems in turn.

3.2 YODIE

As previously discussed, entity disambiguation in YODIE is based on a wide range of features derived from context, Wikipedia, and DBpedia, which are used by a supervised classifier (SVM). Many of the disambiguation features have already been used by other state-of-the-art NEL systems. YODIE, however, also explores a number of novel ideas, focused in particular on better use of the wider context available in some social media content. The most significant developments in the latest version are:

- Contextual similarity has been extended with three further metrics. Each uses a TF/IDF vector space model, calculated over DBpedia content, and the cosine similarity metric. The

first compares mention context with the DBpedia abstract for each candidate. The second adds the value of all textual properties to the candidate vector. The third includes textual content from related entities in the candidate vector.

- Semantic similarity has been extended with an additional metric that calculates relatedness [MW08b] based on links found in Wikipedia, a more comprehensive source of relatedness information than using DBpedia alone.
- Case correction has been introduced, to better handle the often badly-cased content in social media.
- URL expansion has been included, having been demonstrated to provide a significant additional contextual information, useful for disambiguating entities in the case of short social media messages, where highly abbreviated information is often supplemented by a link to a more explanatory web page.
- A support vector machine ML model now replaces the previously used MaxEnt model for disambiguation, having been found significantly superior in identifying nils (i.e. cases where no target entity exists in DBpedia) and spurious mentions.
- A variety of other minor system improvements and additions to the feature set, such as link probability for nil detection and class awareness. Link probability describes the frequency with which a particular textual surface form appears as a link in Wikipedia, and tends to indicate whether it is likely to be a named entity or not. Class awareness information indicates whether the class for the entity as found by ANNIE corresponds to the class of the candidate, where a matching candidate is more likely.
- YODIE uses DBpedia version 3.8 for English language disambiguation and DBpedia version 3.7 for disambiguation of German, Bulgarian, and Hindi texts.

3.3 Evaluation Methodology

The evaluation experiments reported here have been carried out on representative corpora, using well accepted evaluation metrics. It is possible, however, that some of the systems being evaluated here were designed with very different use cases and metrics in mind. For example, systems which aim to cover a very broad range of named entities would appear to produce many spurious NEs, when evaluated on datasets with a more limited NE scope.

In keeping with best practice, we did not tune parameters on the evaluation corpora, neither for YODIE, nor for the other state-of-the-art systems. Nonetheless, since YODIE was trained on both news and social media content (not part of the evaluation corpora), this may give it a certain advantage over other systems. We are more limited in our ability to re-tune competitor systems, so instead a number of heuristic choices were made. These should be borne in mind when interpreting results.

- All systems were used with all parameters set to their default values. For systems which are available via a web API this means we did not specify any request parameters which had a

default setting or explicitly supplied the default settings documented on their web site. Since those default values may not necessarily result into optimal F-measure, it might be possible for the system authors to obtain better evaluation results on our datasets with a different set of parameter values. For example, the default settings for the TagME system are intended to optimize recall and consequently produce spurious results which lowers precision. (Note, however, that the F-measure statistic aims to provide a measure of performance that takes into account the trade-off between precision and recall.)

- Different systems may use different versions of the underlying knowledge bases and versions that are different from the version used in YODIE. Since some URIs may have changed between DBpedia versions, the evaluation results can be influenced by the version of the knowledge base.
- Different URIs may refer to the same knowledge base resource and systems may return any of these URIs for a given entity. Our evaluation system makes an attempt to map all URIs to a single canonical form based on the URL redirect and the sameAs information available from DBpedia, but this may still lead to slight differences, if URIs come from different versions of the knowledge base.
- Some systems do not return DBpedia URIs but instead other identifiers, which can be mapped to DBpedia URIs. For example, AIDA returns YAGO identifiers and Wikipedia URLs and we map the Wikipedia URLs to DBpedia URIs.
- There are many different ways to represent identical URIs (as URIs or IRIs, with or without percent-encoding of certain characters). We map all URIs to the IRI representation used for DBpedia version 3.8.

3.3.1 AIDA

AIDA [HYB⁺11] (<http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/aida/>) is an open source entity linking system developed at the *Max Planck Institut für Informatik*. As recommended on their web page, we did not use the online demo for the evaluation, but deployed their software (available from <https://github.com/yago-naga/aida>) on our own server. All evaluations were done using the version from August 1st, 2014 and used the prepared dataset `AIDA_entity_repository_2014-01-02v7.sql.bz2` available from <http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/aida/downloads/>. Aida was only available for English texts at the time our evaluation was carried out.

3.3.2 Lupedia

Lupedia (<http://lupedia.ontotext.com/>), a free service from Ontotext developed as part of the NoTube project (see <http://notube.tv/things-to-read/deliverables/>), provides automatic entity look-up and disambiguation in text documents, in 8 languages. It works inside a browser or via an API and will automatically identify DBpedia

entities inside a piece of text; however it does not perform any complex text analysis, e.g. discover entities not already present in DBpedia.

3.3.3 DBpedia Spotlight

DBpedia Spotlight [DJHM13] (<http://dbpedia.org/spotlight>) is available both as a free-to-use web service and as open source software and data which can be installed and deployed on one's own servers. We used the free web services in our evaluation: for English language texts the service available at <http://spotlight.dbpedia.org/rest/> and for German language texts the service at <http://de.dbpedia.org/spotlight/rest/>. Spotlight is highly configurable and can perform various annotation tasks, however in this evaluation we only used the entity annotation service with all parameters set to their default values.

3.3.4 TagME

TagME [FS12] is a system developed at the University of Pisa (<http://acube.di.unipi.it/tagme/>). The service can be used via a web demo (<http://tagme.di.unipi.it/>), as well as a RESTful API (see http://tagme.di.unipi.it/tagme_help.html). For our evaluation we used the default settings for all parameters, except one. Since TagME supports tweet analysis differently from longer texts, the `tweet` parameter was set to `true` for the Twitter corpora and to `false` for all other corpora. TagME was only used on the English texts in our evaluation.

3.3.5 TextRazor

TextRazor (<https://www.textrazor.com/>) is a UK startup founded in 2011. It offers a web API for semantic annotation, which can be used for up to 500 requests a day for free.

3.3.6 Zemanta

Zemanta (<http://www.zemanta.com/>) is a Slovenian startup founded in 2007, providing a platform for automatic enrichment of blogs and other online content. The Zemanta API provides the following functionality: entity extraction; related content recommendation; and tag recommendation.

3.4 Corpora

3.4.1 Introduction

An array of evaluation datasets have been selected in order to demonstrate performance on different text types and domains. Since corpora tend to have widely differing properties, the following factors tend to influence performance:

- Text type; for example, news articles or social media content. News articles tend to be well-formed and grammatical, but often have stylizations such as titles in all capitals. Social media content is likely to be less well-formed, possibly poorly capitalized, such as entirely in lower or upper case, and contain abbreviations and jargon. Systems may perform well on one of these types of text, but less well on the other. To fully understand the performance of a system, both must be measured, and the end use case must also be borne in mind when results are interpreted.
- Corpus source; evaluation corpora may have been prepared for different task types. Corpora may have been fully or only partly annotated, meaning that the absence of an annotation may or may not signify that an NE is not present. Corpora may have been chosen to highlight particular abilities on the part of the system; for example, ability to cluster entities. In that case, the corpus may be biased toward a small number of particular entities.
- Quality; corpora may have been automatically annotated with named entities, and employed human annotators only at the stage of selecting the correct referent. Where NEs were automatically detected, a system using the same NER will have an advantage, and others will have a disadvantage, since there is the potential for valid NEs to have been missed by the automated system, making the gold standard incorrect and unfairly penalizing systems that correctly identify that NE. Corpora may also be biased in terms of content. Annotators may have received varied instructions, such that corpora may be annotated in different styles. Where annotation is idiosyncratic, systems performing well may not do so well on other corpora.
- Task assumptions; as stated above, corpora can be prepared with particular tasks in mind, but more broadly, systems may make certain task assumptions. Commercial systems may define success in terms of the number of annotations applied where the referent is accurate and useful. Where a corpus has been prepared with a focus on certain entity types of interest, such systems may be unfairly penalized for adding annotations not present in the gold standard.
- Since our aim is to evaluate entity linking, where the linking is expressed by entity URIs, several technical details about how the URIs were selected for the corpus become relevant. For example, some URIs may be specific to one version of the entity database and not exist in another version. Links can also be expressed by either URIs or IRIs and either version can use various styles of how subsets of characters get encoded and these can also vary between versions. Finally, the original version of the corpus may only contain links which are mappable but not identical to the URIs used in a linking system. In that case the mapping may be incomplete or depend on the version of the knowledge base that contains the mapping relationships.

Additionally, the factors outlined above affect their use as training data. Next we outline the corpora chosen for training and evaluation in these experiments. Evaluation corpora have been selected to allow a comprehensive understanding and comparison of system performance. Training corpora have been selected to provide as much representative data as possible. In introducing the corpora, we bear in mind the points above.

3.4.2 Evaluation Corpora

English

Several English corpora have been selected. English is a common language for comparative evaluation, and so more gold standard data is available. We have also created further data ourselves.

- Tweet data (424 named entities in 397 documents, 0 nils); as part of the project, a gold standard corpus of tweets has been created. These tweets were annotated by domain experts, both at the stage of identifying NEs and choosing the correct DBpedia referents. The Tweets are comprehensively annotated with all person, location, organization and product referents. The annotations are of a high quality, having been viewed by multiple annotators and adjudicated by an expert. A range of topic domains are included, making this a reasonable representation of diverse Twitter content. We reserve half of these tweets, selected at random, for training (as discussed below) and use the remainder for evaluation.
- TAC 2010 data (2228 named entities in 2230 documents, 1273 nils); over a number of years, the Text Analysis Conference (TAC) has been running comparison tasks, including one on named entity linking. Previous years' data are therefore available. Large numbers of documents are included; however, only one gold standard annotation is included in each document, meaning that this corpus cannot be used to assess recall performance. Document type is news articles, that tend to be well-formed, and subject matter is reasonably diverse.
- AIDA data, evaluation set (B) (4485 named entities in 230 documents, 0 nils); the creators of the AIDA system (discussed in the Systems chapter) have made public their dataset. It is a large, fully annotated dataset divided into training, tuning and evaluation sets. We include their evaluation set among our evaluation sets, with training and tuning corpora reserved for training and tuning, as recommended, and discussed in the Corpora chapter. As far as we are aware, this corpus is human-annotated, both at the stage of identifying the entities and linking the correct referents. It has some idiosyncracies; demonyms are included, for example, where other corpora have considered only proper nouns. However, it is a valuable corpus of well-formed news text.
- AIDA EE [HAW14] (9976 named entities in 298 documents, 573 nils) is a corpus available from <http://resources.mpi-inf.mpg.de/yago-naga/aida/download/AIDA-EE.tar.gz>. We added the AIDA EE corpus to our evaluation, but with some reservations. The Stanford NER system has been used to find the NEs, which human annotators then exclusively focused on in linking the correct targets. This gives an advantage to systems based on Stanford NER (AIDA itself included), and penalizes those that aren't. Additionally, the subject matter is biased toward sport news, meaning that systems that happen to perform well on sports news will appear particularly successful, despite that perhaps not being the case on a more mixed corpus. In sports news, country names so often refer to sports teams ("England played Scotland on Thursday"), which is a minority interpretation for a country name. Correctly identifying these may or may not be a strength for a particular system, but ordinarily would not be so extremely important.

German

For German, the News-100 corpus, part of the N3 corpus [RUH⁺14] was used. The version in NIF format available from <https://raw.githubusercontent.com/AKSW/n3-collection/master/News-100.ttl> was converted to GATE format for the evaluation. This corpus consists of 98 news documents with 1619 entities and no annotated nils.

Bulgarian

We used an updated version of the Bulgarian HPSG tree-bank [SOSK04], which was extended by annotations for named entities. Entities were annotated with the resource name of the localized Bulgarian DBpedia entry if it existed, otherwise with the localized Wikipedia page title. Localized resource names and Wikipedia titles were converted to DBpedia IRIs and mapped to english DBpedia URIs (as used by YODIE for the annotation of Bulgarian texts) where possible. The corpus contains 36 news documents (some documents contain the text of more than one news article), 7477 entities, and no nils.

Hindi

As part of the project, we hand-annotated a Hindi evaluation corpus, in order to demonstrate performance in Hindi. This corpus contains a total of 155 annotated tweet documents, containing 199 named entities and no nils.

3.4.3 Training and Tuning Corpora

The success of the YODIE approach to disambiguation depends on a large amount of training data being available. We have compiled a large corpus of training data, and this has enabled us to create a support vector machine that is particularly successful in detecting spurious entities, in addition to providing a good reranking of the candidate list, thus making a large contribution to the success of this state of the art system. In this section, we describe the training data that we have used.

- Previous TAC KBP data: TAC data from the years 2009, 2011 and 2012 was used. Corpora were filtered to ensure no data from the TAC 2010 set was included, since this would compromise the validity of TAC 2010 as an evaluation set. This involved removing a small number of documents. However, the great majority were available for use. Tac 2009 comprises 3688 documents minus the 13 duplicates with TAC 2010, leaving a total of 3675, with around one named entity per document. TAC 2011 comprises 2231 items, of which none are duplicated with TAC 2010. TAC 2012 comprises 2015 items, of which 8 are duplicated with TAC 2010, leaving a total of 2007, again with around one named entity per document.
- AIDA training set: AIDA provide a large training set containing 941 documents and approaching 200 named entities per document on average.
- Tweets: we have created a corpus of fully annotated tweets, which have been reserved for training. This corpus contains 192 tweets, and around 200 named entities.

- AIDA A tuning set: As noted above, AIDA provide a dataset that they suggest should be used for tuning. All YODIE parameter tuning experiments were done using this corpus, e.g. selecting parameters for the SVM disambiguation component.


3.4.4 Tweet Corpus Annotation

Microblog named entity linking (NEL) is a relatively new, underexplored task. Research in this area has focused primarily on *whole-tweet entity linking* (e.g. [ACG⁺12, MWdR12]), also referred to as an “aboutness” task. The whole-tweet NEL task is defined as determining which topics and entities best capture the meaning of a microtext. However, given the shortness of microtext, correct semantic interpretation is often reliant on subtle contextual clues, and needs to be combined with human knowledge. For example, a tweet mentioning iPad makes Apple a relevant entity, because there is the implicit connection between the two. Consequently, entities relevant to a tweet may only be referred to implicitly, without a mention in the tweet’s text. From a corpus annotation perspective, the aboutness task involves identifying relevant entities at whole-document level, skipping the common NER step of determining entity bounds. Both these variants are particularly difficult in microblog genre text (e.g. tweets) [DMAB13b].

Our focus however is on *word level entity linking*, where the task is to disambiguate only the named entities which are mentioned explicitly in the tweet text, by assigning an entity identifier to each named entity mention. However, unlike TAC KBP [JGD⁺10] where only one entity mention per document is disambiguated, we annotate all entity mentions with disambiguated URIs (Unique Reference Identifiers).

NEL Annotation Scheme

Our entity linking annotations are encoded as Mentions, with a start and end offset and an inst feature whose value is a DBpedia URI (see Figure 3.1). They are currently kept separate from the named entity annotations, but the two annotation layers are co-extensive and can easily be merged automatically.



Type	Set	Start	End	Id
Mention	consensus	0	6	89 {inst=http://dbpedia.org/resource/PayPal}
Mention	consensus	8	16	90 {inst=http://dbpedia.org/resource/Coinstar}

Figure 3.1: Word Level Entity Linking Annotations, shown in GATE

We chose DBpedia [BLK⁺09] as the target entity linking database, due to its good coverage of named entities, its frequent updates, and available mappings to other Linked Open Data resources, such as YAGO and Freebase.

Task Design and Data Preparation

NEL is essentially a classification task, where the goal is to choose amongst one of the possible entity targets from the knowledge base or NIL (no target entity), in cases where no such entity

exists. The latter case is quite common in tweets, where people often refer to friends and family, for example. An added problem, however, is that highly ambiguous entity mentions (e.g. Paris), could have tens or even over a hundred possible target entities. Since showing so many options to a human is not feasible, instead, during data preparation, candidate entity URIs are ranked according to their Wikipedia commonness score [MW08a] and only the top 8 are retained and shown, in addition to NIL (which we called “none of the above”) and “not an entity” (to allow for errors in the automatic pre-processing). We chose to show at most 8 entity candidates, following a small-scale pilot with NLP experts, which gave us feedback.

In order to overcome the problem that the correct candidate entity could have been present in DBpedia, but filtered out due to low occurrence in Wikipedia, we allowed NLP experts, who are also familiar with DBpedia, to search and identify the correct entity URI in such cases.

As can be seen, the key data preparation step is the generation of candidate entity URIs. Even though error prone, candidate entity selection against DBpedia needs to be carried out automatically, since the latest English DBpedia contains 832,000 persons, 639,000 places, and 209,000 organisations, out of the 4 million DBpedia URIs in total.

Relying purely on looking up exact matching labels in DBpedia is not sufficient, since entities are often referred to by acronyms, nicknames, and shortened names (e.g. surnames like Obama or Snowden). Instead, we match the string of the named entity mention in the document (annotated already in the corpus) against the values of the *rdf:label*, *foaf:name* and several other similar annotation properties, for all instances of the *dbpedia-ont:Person*, *dbpedia-ont:Organisation* and *dbpedia-ont:Place* classes in DBpedia. Acronyms and shorter ways of referring to DBpedia entity URIs are collected also from the Wikipedia anchor texts, that point to the respective Wikipedia page.¹

Lastly, we had to choose the size of the context, shown to the annotators to help with the disambiguation of the entity mention. We experimented with showing the sentence where the mention appears, but this was not sufficient. Therefore, we showed the entire tweet text and any web links within. For longer documents, it would make sense to show at least 1 preceding and 1 following sentence, or even the containing paragraph, space permitting.

The NEL Annotation Interface

We designed a CrowdFlower-based user interface (see Figure 3.2), which showed the text of the tweet, any URL links contained therein, and a set of candidate targets from DBpedia. The instructions encouraged the annotators to click on the URL links from the tweet, in order to gain addition context and thus ensure that the correct DBpedia URI is chosen.

Candidate entity meanings were shown in random order, using the text from the corresponding DBpedia abstracts (where available) or the actual DBpedia URI otherwise.

In addition, the options “none of the above” and “not an entity” were added, to allow the annotators to indicate that this entity mention has no corresponding DBpedia URI (none of the above) or that the highlighted text is not an entity. We also added a third option “cannot decide”, so the annotators could indicate that the context did not provide sufficient information to reliably

¹There is a 1-to-1 mapping between each DBpedia URI and the corresponding Wikipedia page, which makes it possible to treat Wikipedia as a large corpus, human annotated with DBpedia URIs.

The screenshot shows a web interface for a task titled "Entity Disambiguation Task". At the top, there's a header with a logo, "Work mode", "11 tasks completed", "1 cents per task", and a user profile "Suman Aswani". A timer indicates "29:19 left for this task". Below the title is an "Instructions" button. The main content area displays a tweet snippet: "Exclusive : Rep . **Steve King** on ObamaCare , Tea Party , and Constitution Day : The inclusion of the Tenth Amendment in ... http://bit.ly/cYITA8". It lists "URLs in the tweet:" as "http://bit.ly/cYITA8". The task question is "Which of the descriptions below describes 'Steve King' best?". There are five radio button options:

- ☐ Steven Arnold Steve King (born May 28, 1949) is the U.S. Representative for Iowa's 5th congressional district, serving since 2003. He is a member of the Republican Party. The district is located in the western part of the state and includes Sioux City and Council Bluffs. .
- ☐ Steve King is a legislator in the U.S. state of Colorado. Elected to the Colorado House of Representatives as a Republican in 2006, King represents House District 54, encompassing southern Mesa County and western Delta County, Colorado. .
- ☐ For other people named Steve King, see Stephen King (disambiguation). Template:Infobox gridiron football person George Stephen King (born June 10, 1951) is a former American football linebacker in the National Football League. He graduated from Quinton high school in Quinton, Oklahoma in 1969. He then played for The University of Tulsa. He also played nine seasons for the New England Patriots. .
- ☐ Stephen F. King (1842-1895) was an American professional baseball player who played in the National Association as an outfielder for the 1871-1872 Troy Haymakers. .
- ☐ None of the above
- ☐ Not an Entity
- ☐ Cannot decide

Figure 3.2: The NEL Annotation Interface

disambiguate the entity mention.

It must be noted that even though the annotation interface was hosted by CrowdFlower, the actual tweet annotation was carried out by domain experts from HFPL and USFD and not paid-for workers.

The Annotation Process

We chose a random set of 177 entity mentions for the expert-sourcing NEL pilot and generated candidate URIs for them. Each entity mention was disambiguated by a random set of three NLP experts, using our NEL annotation interface. We had 10 volunteer experts in total for this pilot.

Annotations for which no clear decision was made were adjudicated by a fourth expert, who had not previously seen the tweets.

As each entity annotation was disambiguated by three NLP volunteers, we determined agreement by measuring the proportion of annotations on which all three made the same choice. Out of the resulting 531 judgements, unanimous inter-annotator agreement occurred for 89% of entities.

The resulting expert-sourced dataset consisted of 172 microblog texts, containing entity mentions and their assigned DBpedia URIs.

Given the high level of inter-annotator agreement, we then used two annotators per tweet, with a third adjudicator reconciling the differences between the first two. This was done in order to enable us to annotate more tweets at a reasonable time.

The resulting English tweet NEL corpus was then split into the training and evaluation portions

described in the previous section.

3.5 Metrics

3.5.1 Introduction

The named entity linking task overarches a number of subtasks, and evaluation varies depending on what a system aims to achieve.

- Named entity detection (NED) involves locating named entities in text, but not identifying their referents. Many systems exist that perform only NER, for example, ANNIE and Stanford NER.
- Named entity linking (NEL) involves identifying the referent of a named entity, but not the task of finding the named entity in the first place. The TAC KBP task evaluates this subtask only.
- Named entity recognition and disambiguation (NERD, or just NER) describes the overarching task of both finding and disambiguating named entities in text. Linking to the DBpedia referent is equivalent to disambiguation since DBpedia represents the knowledge base by which the referent is identified. This is the task that YODIE performs, and constitutes the more useful task in practical terms.

As a NERD system, LODIE faces the challenge not only of correctly identifying referents, but also not overgenerating—that is, not hypothesizing named entities to exist where there are none. However, it is also an important metric of performance how well YODIE, in comparison with the other systems, is able to perform NEL. Therefore, we use a variety of metrics.

3.5.2 NERD Metrics

Precision describes the proportion of named entities found by the system that are correct. That is, of all the times that the system identified a particular entity as being referred to in a particular location, how often was it correct? This statistic penalizes overgeneration—although a system that finds many entities will be more likely not to miss one, it will generate more wrong answers and thus have a lower precision.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

Recall describes the proportion of times that the system correctly found a named entity that is present. That is, of all the named entities in the text, how often did the system find it? This statistic penalizes undergeneration—although a system that only identifies named entities it is sure of will have a high precision, recall will be low.

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

Since precision and recall trade off against each other to a great extent, a combined metric provides a more comparable statistic. The F-measure offers this, and can be tuned for situations in which precision (or recall) is more important. In our task, we have no argument for why precision or recall would be more important, so we use the metric F1.

$$F1 = 2 \frac{Precision * Recall}{Precision + Recall}$$

Where an entity is located and correctly linked but the location is not exactly correct, we consider this to be a partially correct answer. Overlapping but not identical spans often occur in the NERD task because there can be room for opinion about where a named entity starts and ends. For example, in “The White House”, is “the” part of the named entity or not? In “Prime Minister David Cameron”, is “Prime Minister” part of a single named entity referring to David Cameron, or is it a separate named entity referring to a political position in the United Kingdom? We consider it in the spirit of the task to count partially correct answers as being correct, because span errors are of little consequence in the utility of a system. Responses in the correct location but linked to the wrong entity are, of course, not counted at all. For this reason, we use variants on the above metrics known as “lenient” metrics, which calculate precision and recall as follows:

$$Precision = \frac{TruePositives + Partial}{TruePositives + Partial + FalsePositives}$$

$$Recall = \frac{TruePositives + Partial}{TruePositives + Partial + FalseNegatives}$$

3.5.3 Entity Linking Metrics

The separate task of evaluating the extent to which a system, given the correct location, is able to link the right referent requires a different metric. Accuracy describes the proportion of named entities in the document that were correctly linked by the system. It is a similar statistic to recall except that nils are handled differently, in that they are not excluded from the calculation. This is in the spirit of the TAC KBP task, where a large number of nil entities are given in the evaluation data (named entities that don’t have a referent in DBpedia) and correctly identifying these nils is considered an important part of the task. In the statistics given above, an entity in an evaluation corpus that has not been linked, but instead has been annotated as a nil, is treated as though it is not there. For accuracy, however, nils that have not been annotated by a system or have been annotated as a nil are treated as being correct.

$$Accuracy = \frac{Correct}{TotalNamedEntities}$$

As above, we present, for all systems, a lenient accuracy. Particularly in the case of accuracy, the systems are not being evaluated for their ability to locate the named entities, but for their ability to correctly disambiguate them. Therefore, we do not penalize for span variations. In the case that the system creates two named entities overlapping the key span (for example, one overlaps the start of the key span and the other, the end) we evaluate only the first.

$$\textit{LenientAccuracy} = \frac{\textit{Correct} + \textit{Partial}}{\textit{TotalNamedEntities}}$$

Depending on the corpus, the impact of nils varies widely. For corpora with many nils, accuracy may be quite different to recall.

Chapter 4

Results

4.1 English

Results are presented for YODIE and six state-of-the-art systems on four different corpora for the English language.

4.1.1 397 Tweets Test Corpus

Table 4.1 shows results for the six state-of-the-art systems contrasted with YODIE on the tweets evaluation corpus. Results on the the tweets corpus are particularly relevant since social media is a focus in TrendMiner. Results show that for several of the systems, high accuracies are achieved at the expense of precision. Thus, for example, TagME’s particularly high accuracy comes with a low F1. For all-round performers, the most impressive results come from Zemanta and YODIE. Results are comparable for the two systems on this corpus. In more detail, YODIE has slightly better precision and F-score than Zemanta, but the latter has higher recall and accuracy. However, results on other corpora demonstrate that YODIE achieves a superior all-round performance on a variety of different text types.

System	Prec	Rec	F1	Acc
YODIE	0.51	0.54	0.53	0.54
Aida	0.59	0.38	0.46	0.38
Lupedia	0.50	0.24	0.32	0.24
Spotlight	0.09	0.51	0.15	0.46
TagMe	0.10	0.67	0.17	0.67
TextRazor	0.19	0.44	0.26	0.44
Zemanta	0.48	0.56	0.52	0.56

Table 4.1: Comparison of performance on the 397 tweets corpus

System	Prec	Rec	F1	Acc
YODIE	0.62	0.65	0.64	0.65
Aida	0.70	0.74	0.72	0.74
Lupedia	0.58	0.31	0.40	0.31
Spotlight	0.22	0.49	0.31	0.49
TagMe	0.18	0.45	0.26	0.45
TextRazor	0.35	0.58	0.43	0.58
Zemanta	0.51	0.29	0.37	0.29

Table 4.2: Comparison of performance on the AIDA B news corpus

System	Prec	Rec	F1	Acc
YODIE	n/a	n/a	n/a	0.57
Lupedia	n/a	n/a	n/a	0.57
Spotlight	n/a	n/a	n/a	0.55
TagMe	n/a	n/a	n/a	0.40
TextRazor	n/a	n/a	n/a	0.60
Zemanta	n/a	n/a	n/a	0.53

Table 4.3: Comparison of performance on the TAC 2010 news corpus

4.1.2 AIDA B (AIDA test set)

AIDA B, as discussed earlier, is a news corpus, and thus poses slightly different challenges to the tweets corpus. YODIE achieves a convincingly superior performance on this test set (see Table 4.2), with wide margins separating performance from the nearest competitor. The only exception is the AIDA system itself, which shows a particularly good performance on its own corpus that is not replicated on the tweets corpus. Note also that Zemanta do not replicate their success on the tweets corpus on this type of data.

4.1.3 TAC 2010

The TAC 2010 corpus is idiosyncratic, as discussed earlier, in that it can only be used to evaluate the disambiguation part of the system, hence only accuracy is given. It also contains a large number of nils.

As can be seen in Table 4.3, YODIE demonstrates a solid performance, coming in joint second place. The success of TextRazor on this corpus and task perhaps follows from their skew toward recall, and shows how for a corpus that can only be used to assess accuracy, systems that favour recall over precision have an advantage because there is no penalty for a low precision. Succeeding at this kind of task using a full NERD system is heavily influenced by tuning.

Results on this corpus demonstrate the solid performance achievable by YODIE without compromising on other corpus types. We were not able to obtain results for Aida using the same default parameters as elsewhere on this corpus because we ran into out of memory problems for some documents.

System	Prec	Rec	F1	Acc
YODIE	0.55	0.58	0.57	0.58
Aida	0.69	0.78	0.73	0.74
Lupedia	0.53	0.32	0.40	0.34
Spotlight	0.19	0.56	0.29	0.57
TagMe	0.09	0.37	0.15	0.38
TextRazor	0.27	0.62	0.37	0.63
Zemanta	0.49	0.23	0.32	0.27

Table 4.4: Comparison of performance on the AIDA EE news corpus

System	Prec	Rec	F1	Acc
YODIE	0.50	0.43	0.46	0.43
Lupedia	0.38	0.26	0.31	0.26
Spotlight	0.54	0.43	0.48	0.43
TextRazor	0.32	0.51	0.39	0.51

Table 4.5: Comparison of performance on the German News-100 corpus

4.1.4 AIDA EE

On the AIDA EE corpus, YODIE shows the most solid all-round performance from among the fairly compared systems. Recall that the Aida system uses the same NER system (Stanford NER), as was used in annotating the corpus, so this gives them an advantage, which makes it impossible to compare the result directly with the other systems. Their result here is included for reference only.

Basides, as noted above, the AIDA EE corpus is quite heavily skewed towards sports, which is not a domain of importance to TrendMiner.

Of the remainder, whilst TextRazor has the highest accuracy, as we have seen previously on other corpora, this is achieved at the expense of precision, and hence the F1 score is low. Zemanta, again, do not show the high performance they achieve on tweets. YODIE not only has a high accuracy, but has the highest F1 by a considerable margin.

4.2 German

Several state-of-the-art NEL systems exist for the German language, allowing for a reasonably comprehensive evaluation. YODIE comes in joint second place of four for accuracy, after TextRazor, as shown in table 4.5, and second place to DBpedia Spotlight for F1.

4.3 Bulgarian

We were able to compare system performance in Bulgarian against only one other NEL system, Lupedia, that also annotates Bulgarian. Therefore we are able to see that our accuracy is ahead of

System	Prec	Rec	F1	Acc
YODIE	0.38	0.41	0.39	0.41
Lupedia	0.66	0.30	0.41	0.30

Table 4.6: Comparison of performance on the Bulgarian corpus

System	Prec	Rec	F1	Acc
YODIE	0.31	0.27	0.29	0.27

Table 4.7: Performance on the Hindi tweet corpus

Lupedia by a good margin, and our F1 is very close to equalling Lupedia, with only two percentage points between the two results, as shown in table 4.6.

4.4 Hindi

Although there are no other systems annotating Hindi against which we can compare our result, we can gauge YODIE’s performance on Hindi by comparing roughly against the English tweet results, since the Hindi corpus is also tweets. As shown in table 4.7, midrange performance is achieved, with scores that would have come in ahead of or equal to four of the six English systems compared above for F1 and two for accuracy.

The challenges faced by the Hindi YODIE is the lack of a mature NER component, no adaptation to tweets specifically, the lower coverage of the Hindi DBpedia, and the issues faced by the automatic transliteration process, which is used in an attempt to broaden recall via the English DBpedia.

4.5 Discussion

We have demonstrated state of the art performance of our YODIE system on the challenging task of combined named entity recognition and disambiguation. On the AIDA B dataset, YODIE performance is ahead of competing systems with the exception of the AIDA system. On the AIDA EE data, YODIE outperforms competing systems—although TextRazor comes in slightly ahead on accuracy, YODIE has a far superior F1 indicating a better performance balance for the TrendMiner use cases.

On the tweets corpus, YODIE substantially outperforms all competitors with the exception of Zemanta, who achieve a very good result on this particular type of data. Nevertheless, it should be noted that YODIE’s performance is similar to Zemanta’s, while the latter performs much worse than YODIE on the other corpora.

To summarise, our evaluation experiments have demonstrated that YODIE has the most solid all-round result on a range of different textual genres and domains, performing consistently to a high standard.

We attribute this success to several factors:

- YODIE’s basis in the well-established GATE architecture allowed us to leverage a wider community effort in rapidly integrating and implementing a variety of different entity linking features, making YODIE a particularly comprehensive system. We were able to rapidly include features such as entity class, as determined by the ANNIE information extraction system, and part of speech information, in order to create a rich feature set for disambiguation.
- YODIE includes a wide variety of features at the disambiguation stage, as detailed in D2.2.2 and in chapter 3
- A “more is more” attitude to feature generation was found advantageous given our choice of machine learning algorithm at the disambiguation stage. This synergy resulted in a solid overall performance.

Chapter 5

Combining and Evaluating Polarity Lexicons

5.1 Introduction

Despite the rapid growth of the sentiment mining area, there is a lack of gold-standard corpora which can be used to train supervised models, particularly for languages other than English. Consequently, many algorithms rely on sentiment lexicons, which provide prior knowledge about which lexical items might indicate opinionated language. Such lexicons can be used directly to define features in a classifier, or can be combined with a bootstrapping approach.

However, when presented with a number of overlapping and potentially contradictory sentiment lexicons, many machine learning techniques break down, and we therefore require a way to merge them into a single resource - or else a researcher must choose between resources, and we are left with a leaky pipeline between resource creation and application. We review methods for combining sources of information in section 5.2, and then describe four German sentiment lexicons in section 5.3.

To merge these resources, we first want to make them match as closely as possible, and then deal with the differences that remain. We deal with the first step in section 5.4, describing how to align the polarity scores in different lexicons so that they can be directly compared. Then in section 5.5, we describe how to combine these scores together.

We report results in section 5.6, including evaluation against a small annotated corpus, where our merged resource outperforms both the original resources and also a majority vote baseline. Finally, we discuss distribution of our resource in section 5.7 and propose future work in section 5.8.

5.2 Related Work

A general problem is how to deal with missing data - in our case, we cannot expect every word to appear in every lexicon. [SG02] review techniques to deal with missing data, and recommend two approaches: maximum likelihood estimation and Bayesian multiple imputation. The latter is a Monte Carlo method, helpful when the marginal probability distribution cannot be calculated

Lexicon	# Entries
C&K	8714
PolarityClues	9228
SentiWS	1896
SentiSpin	95572
SentiMerge	96918

Table 5.1: Comparison of lexicon sizes

analytically. The probabilistic model presented in section 5.5.1 is straightforward enough for marginal probabilities to be calculated directly, and we employ maximum likelihood estimation for this reason.

A second problem is how to combine multiple sources of information, which possibly conflict, and where some sources are more reliable than others. This becomes particularly challenging in the case when no gold-standard data exists, and so the sources can not be evaluated directly. [RYZ⁺10] discusses this problem from the point of view of crowdsourcing, where there are multiple expert views and no certain ground truth - but we can equally apply this in the context of sentiment analysis, viewing each source as an expert. However, unlike their approach, our algorithm does not directly produce a classifier, but rather a newly labelled resource.

Confronted with a multiplicity of data sources, some researchers have opted to link resources together [EKG13]. Indeed, the lexicons we consider in section 5.3 have already been compiled into a common format by [DK14]. However, while linking resources makes it easier to access a larger amount of data, it does not solve the problem of how best to process it.

To the best of our knowledge, there has not been a previous attempt to use a probabilistic model to merge a number of sentiment lexicons into a single resource.

5.3 Data Sources

In the following subsections, we first describe four existing sentiment lexicons for German. These four lexicons represent the data we have merged into a single resource, with a size comparison given in table 5.1, where we count the number of distinct lemmas, not considering parts of speech. Finally, in section 5.3.5, we describe the manually annotated MLSA corpus, which we use for evaluation.

5.3.1 Clematide and Klenner

[CK10] manually curated a lexicon¹ of around 8000 words, based on the synsets in *GermaNet*, a *WordNet*-like database [HF97]. A semi-automatic approach was used to extend the lexicon, first generating candidate polar words by searching in a corpus for coordination with known polar words, and then presenting these words to human annotators. We will refer to this resource as the C&K lexicon.

¹<http://bics.sentimental.li/index.php/downloads>

5.3.2 SentimentWortschatz

[RQH10] compiled a sentiment lexicon² from three data sources: a German translation of [SDS66]’s *General Inquirer* lexicon, a set of rated product reviews, and a German collocation dictionary. At this stage, words have binary polarity: positive or negative. To assign polarity weights, they use a corpus to calculate the mutual information of a target word with a small set of seed words.

5.3.3 GermanSentiSpin

[TIO05] produced SentiSpin, a sentiment lexicon for English. It is so named because it applies the Ising Model of electron spins. The lexicon is modelled as an undirected graph, with each word type represented by a single node. A dictionary is used to define edges: two nodes are connected if one word appears in the other’s definition. Each word is modelled as having either positive or negative sentiment, analogous to electrons being spin up or spin down. An energy function is defined across the whole graph, which prefers words to have the same sentiment if they are linked together. By using a small seed set of words which are manually assigned positive or negative sentiment, this energy function allows us to propagate sentiment across the entire graph, assigning each word a real-valued sentiment score in the interval $[-1, 1]$.

[Wal10b] translated the SentiSpin resource into German³ using an online dictionary, taking at most three translations of each English word.

5.3.4 GermanPolarityClues

[Wal10a] utilised automatic translations of two English resources: the SentiSpin lexicon, described in section 5.3.3 above; and the Subjectivity Clues lexicon, a manually annotated lexicon produced by [WWH05]. The sentiment orientations of the German translations were then manually assessed and corrected where necessary, to produce a new resource.⁴

5.3.5 MLSA

To evaluate a sentiment lexicon, separately from the general task of judging the sentiment of an entire sentence, we relied on the MLSA (Multi-Layered reference corpus for German Sentiment Analysis). This corpus was produced by [CGK⁺12], independently of the above four lexicons, and consists of 270 sentences annotated at three levels of granularity. In the first layer, annotators judged the sentiment of whole sentences; in the second layer, the sentiment of words and phrases; and finally in the third layer, they produced a FrameNet-like analysis of each sentence. The third layer also includes lemmas, parts of speech, and a syntactic parse.

We extracted the sentiment judgements of individual words from the second layer, using the majority judgement of the three annotators. Each token was mapped to its lemmatised form and part of speech, using the information in the third layer. In some cases, the lemma was listed as

²<http://asv.informatik.uni-leipzig.de/download/sentiws.html>

³<http://www.ulliwaltinger.de/sentiment>

⁴<http://www.ulliwaltinger.de/sentiment>

Lemma, POS	vergöttern, V
C&K	1.000
PolarityClues	0.333
SentiWS	0.004
SentiSpin	0.245

Table 5.2: An example lemma, labelled with polarity strengths from each data source

ambiguous or unknown, and in these cases, we manually added the correct lemma. Additionally, we changed the annotation of nominalised verbs from nouns to verbs, to match the lexical entries. Finally, we kept all content words (nouns, verbs, and adjectives) to form a set of test data. In total, there were 1001 distinct lemma types, and 1424 tokens. Of these, 378 tokens were annotated as having positive polarity, and 399 as negative.

5.4 Normalising Scores

By considering positive polarity as a positive real number, and negative polarity as a negative real number, all of the four data sources give polarity scores between -1 and 1 . However, we cannot assume that the values directly correspond to one another. For example, does a 0.5 in one source mean the same thing in another? An example of the kind of data we are trying to combine is given in table 5.2, and we can see that the polarity strengths vary wildly between the sources.

The simplest model is to rescale scores linearly, i.e. for each source, we multiply all of its scores by a constant factor. Intuitively, the factors should be chosen to harmonise the values - a source with large scores should have them made smaller, and a source with small scores should have them made larger.

5.4.1 Linear Rescaling for Two Sources

To exemplify our method, we first restrict ourselves to the simpler case of only dealing with two lexicons. Note that when trying to determine the normalisation factors, we only consider words in the overlap between the two; otherwise, we would introduce a bias according to what words are considered in each source - it is only in the overlap that we can compare them. However, once these factors have been determined, we can use them to rescale the scores across the entire lexicon, including items that only appear in one source.

We consider lemmas with their parts of speech, so that the same orthographic word with two possible parts of speech is treated as two independent lexical entries, in all of the following calculations. However, we do not distinguish homophonous or polysemous lemmas within the same part of speech, since none of our data sources provided different sentiment scores for distinct senses.

For each word i , let u_i and v_i be the polarity scores for the two sources. We would like to find positive real values λ and μ to rescale these to λu_i and μv_i respectively, minimising the loss function $\sum_i (\lambda u_i - \mu v_i)^2$. Intuitively, we are trying to rescale the sources so that the scores are as similar as possible. The loss function is trivially minimised when $\lambda = \mu = 0$, since reducing

the sizes of the scores also reduces their difference. Hence, we can introduce the constraint that $\lambda\mu = 1$, so that we cannot simultaneously make the values smaller in both sources. We would then like to minimise:

$$\sum_i \left(\lambda u_i - \frac{1}{\lambda} v_i \right)^2 = |u|^2 \lambda^2 - 2u.v + |v|^2 \lambda^{-2}$$

Note that we use vector notation, so that $|u|^2 = \sum_i u_i^2$. Differentiating this with respect to λ , we get:

$$2\lambda |u|^2 - 2|v|^2 \lambda^{-3} = 0 \quad \Rightarrow \quad \lambda = \frac{\sqrt{|v|}}{\sqrt{|u|}}$$

However, observe that we are free to multiply both λ and μ by a constant factor, since this doesn't affect the relationship between the two sources, only the overall size of the polarity values. By dividing by $\sqrt{\frac{|u||v|}{n}}$, we derive the simpler expressions $\lambda = \sqrt{n} |u|^{-1}$ and $\mu = \sqrt{n} |v|^{-1}$, i.e. we should divide by the root mean square. In other words, after normalising, the average squared polarity value is 1 for both sources.⁵

5.4.2 Rescaling for Multiple Sources

For multiple sources, the above method needs tweaking. Although we could use the overlap between all sources, this could potentially be much smaller than the overlap between any two sources, introducing data sparsity and making the method susceptible to noise. In the given data, 10749 lexical items appear in at least two sources, but only 1205 appear in all four. We would like to exploit this extra information, but the missing data means that methods such as linear regression cannot be applied.

A simple solution is to calculate the root mean square values for each pair of sources, and then average these values for each source. These averaged values define normalisation factors, as a compromise between the various sources.

5.4.3 Unspecified scores

Some lexical items in the PolarityClues dataset were not assigned a numerical score, only a polarity direction. In these cases, the task is not to normalise the score, but to assign one. To do this, we can first normalise the scores of all other words, as described above. Then, we can consider the words without scores, and calculate the root mean square polarity of these words in the other sources, and assign them this value, either positive or negative.

⁵In most sentiment lexicons, polarity strengths are at most 1. This will no longer be true after this normalisation.

5.5 Combining Scores

Now that we have normalised scores, we need to calculate a combined value. Here, we take a Bayesian approach, where we assume that there is a latent “true” polarity value, and each source is an observation of this value, plus some noise.

5.5.1 Gaussian Model

A simple model is to assume that we have a prior distribution of polarity values across the vocabulary, distributed normally. If we further assume that a language is on average neither positive nor negative, then this distribution has mean 0. We denote the variance as σ^2 . Each source independently introduces a linear error term, which we also model with a normal distribution: errors from source a are distributed with mean 0 and standard deviation σ_a^2 , which varies according to the source.⁶

5.5.2 Hyperparameter Selection

If we observe a subset $S = \{a_1, \dots, a_n\}$ of the sources, the marginal distribution of the observations will be normally distributed, with mean 0 and covariance matrix as shown below. If the error variances σ_a^2 are small compared to the background variance σ^2 , then this implies a strong correlation between the observations.

$$\begin{pmatrix} \sigma^2 + \sigma_{a_1}^2 & \sigma^2 & \dots & \sigma^2 \\ \sigma^2 & \sigma^2 + \sigma_{a_2}^2 & \dots & \sigma^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma^2 & \sigma^2 & \dots & \sigma^2 + \sigma_{a_n}^2 \end{pmatrix}$$

To choose the values for σ^2 and σ_a^2 , we can aim to maximise the likelihood of the observations, i.e. maximise the value of the above marginal distributions at the observed points. This is in line with [SG02]’s recommendations. Such an optimisation problem can be dealt with using existing software, such as included in the SciPy⁷ package for Python.

5.5.3 Inference

Given a model as above (whether or not the hyperparameters have been optimised), we can calculate the posterior distribution of polarity values, given the observations x_{a_i} . This again turns out to be normally distributed, with mean $\hat{\mu}$ and variance $\hat{\sigma}^2$ given by:

$$\hat{\mu} = \frac{\sum \sigma_{a_i}^{-2} x_{a_i}}{\sigma^{-2} + \sum \sigma_{a_i}^{-2}}$$

⁶Because of the independence assumptions, this model can alternatively be viewed as a Markov Network, where we have one node to represent the latent true polarity strengths, four nodes to represent observations from each source, and five nodes to represent the hyperparameters (variances)

⁷<http://www.scipy.org>

$$\hat{\sigma}^{-2} = \sigma^{-2} + \sum \sigma_{a_i}^{-2}$$

The mean is almost a weighted average of the observed polarity values, where each source has weight σ_a^{-2} . However, there is an additional term σ^{-2} in the denominator - this means we can interpret this as a weighted average if we add an additional polarity value 0, with weight σ^{-2} . This additional term corresponds to the prior.

The weights for each source intuitively mean that we trust sources more if they have less noise. The extra 0 term from the prior means that we interpret the observations conservatively, skewing values towards 0 when there are fewer observations. For example, if all sources give a large positive polarity value, we can be reasonably certain that the true value is also large and positive, but if we only have data from one source, then we are less certain if this is true - our estimate $\hat{\mu}$ is correspondingly smaller, and the posterior variance $\hat{\sigma}^2$ correspondingly larger.

5.6 Experiments and Results

5.6.1 Parameter Values

The root mean square sentiment values for the sources were: C&K 0.845; PolarityClues 0.608; SentiWS 0.267; and SentiSpin 0.560. We can see that there is a large discrepancy between the sizes of the scores used, with SentiWS having the smallest of all. It is precisely for this reason that we need to normalise the scores.

The optimal variances calculated during hyperparameter selection (section 5.5.2) were: prior 0.528; C&K 0.328; PolarityClues 0.317; SentiWS 0.446; and SentiSpin 0.609. These values correlate with our intuition: C&K and PolarityClues have been hand-crafted, and have smaller error variances; SentiWS was manually finalised, and has a larger error; while finally SentiSpin was automatically generated, and has the largest error of all, larger in fact than the variance in the prior. We would expect the polarity values from a hand-crafted source to be more accurate, and this appears to be justified by our analysis.

5.6.2 Experimental Setup

The MLSA data (see section 5.3.5) consists of discrete polarity judgements - a word is positive, negative, or neutral, but nothing in between.⁸ To allow direct evaluation against such a resource, we need to discretise the continuous range of polarity values; i.e. if the polarity value is above some positive threshold, we judge it to be positive; if it is below a negative threshold, negative; and if it is between the two thresholds, neutral. To choose this threshold before evaluation, we calculated a Gaussian kernel density estimate of the polarity values in the entire lexicon, as shown in figure 5.1. There is a large density near 0, reflecting that the bulk of the vocabulary is not strongly polar; indeed, so that the density of polar items is clearly visible, we have chosen a scale that forces this bulk to go off the top of the chart. The high density stops at around ± 0.23 , and we have accordingly set this as our threshold.

⁸The annotation scheme also allows a further three labels: *intensifier*, *diminisher*, and *shifter*. While this information is useful, we treat these values as neutral in our evaluation, since we are only concerned with words that have an inherent positive or negative polarity.

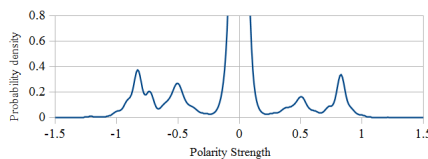


Figure 5.1: Gaussian kernel density estimate

Lexicon	Precision	Recall	F-score
C&K	0.754	0.733	0.743
PolarityClues	0.705	0.564	0.626
SentiWS	0.803	0.513	0.621
SentiSpin	0.557	0.668	0.607
majority vote	0.548	0.898	0.679
SentiMerge	0.708	0.815	0.757

Table 5.3: Performance on MLSA, macro-averaged

We compared the merged resource to each of the original lexicons, as well as a “majority vote” baseline which represents an alternative method to combine lexicons. This baseline involves considering the polarity judgements of each lexicon (*positive*, *negative*, or *neutral*), and taking the most common answer. To break ties, we took the first answer when consulting the lexicons in the following order, reflecting their reliability: C&K, PolarityClues, SentiWS, SentiSpin.

For the automatically derived resources, we can introduce a threshold as we did for SentiMerge. However, to make these baselines as competitive as possible, we optimised them on the test data, rather than choosing them in advance. They were chosen to maximise the macro-averaged f-score. For SentiWS, the threshold was 0, and for SentiSpin, 0.02.

Note that a perfect score would be impossible to achieve, since 31 lemmas were annotated with more than polarity type. These cases generally involve polysemous words which could be interpreted with different polarities depending on the context. Indeed, two words appeared with all three labels: *Spannung* (tension) and *Widerstand* (resistance). In a political context, interpreting *Widerstand* as positive or negative depends very much on whose side you support. In such cases, a greater context is necessary to decide on polarity, and a lexicon simply cannot suffice.

5.6.3 Evaluation on MLSA

We calculated precision, recall, and f-score (the harmonic mean of precision and recall) for both positive and negative polarity. We report the average of these two scores in 5.3. We can see that in terms of f-score, SentiMerge outperforms all four data sources, as well as the majority vote. In applications where either precision or recall is deemed to be more important, it would be possible to adjust the threshold accordingly. Indeed, by dropping the threshold to zero, we achieve recall of 0.894, competitive with the majority vote method; and by increasing the threshold to 0.4, we achieve precision of 0.755, competitive with the C&K lexicon. Furthermore, in this latter case, the f-score also increases to 0.760. We do not report this figure in the table above because it would not be possible to predict such a judicious choice of threshold without peeking at the test data.

Nonetheless, this demonstrates that our method is robust to changes in parameter settings.

The majority vote method performs considerably worse than SentiMerge, at least in terms of f-score. Indeed, it actually performs worse than the C&K lexicon, with noticeably lower precision. This finding is consistent with the results of [RYZ⁺10], who argue against using majority voting, and who also find that it performs poorly.

The C&K lexicon achieves almost the same level of performance as SentiMerge, so it is reasonable to ask if there is any point in building a merged lexicon at all. We believe there are two good reasons for doing this. Firstly, although the C&K lexicon may be the most accurate, it is also small, especially compared to SentiSpin. SentiMerge thus manages to exploit the complementary nature of the different lexicons, achieving the broad coverage of SentiSpin, but maintaining the precision of the C&K lexicon for the most important lexical items.

Secondly, SentiMerge can provide much more accurate values for polarity strength than any human-annotated resource can. As [CK10] show, inter-annotator agreement for polarity strength is low, even when agreement for polarity direction is high. Nonetheless, some notion of polarity strength can still be helpful in computational applications. To demonstrate this, we calculated the precision, recall, and f-scores again, but weighting each answer as a function of the distance from the estimated polarity strength to the threshold. With this weighted approach, we get a macro-averaged f-score of 0.852. This is considerably higher than the results given in table 5.3, which demonstrates that the polarity scores in SentiMerge are useful as a measure of classification certainty.

5.6.4 Manual Inspection

In cases where all sources agree on whether a word is positive or negative, our algorithm simply serves to assign a more accurate polarity strength. So, it is more interesting to consider those cases where the sources disagree on polarity direction. Out of the 1205 lexemes for which we have data from all four sources, only 22 differ between SentiMerge and the C&K lexicon, and only 16 differ between SentiMerge and PolarityClues. One example is *Beschwichtigung* (appeasement). Here we can see the problem with trying to assign a single numeric value to polarity - in a political context, *Beschwichtigung* could be interpreted either as positive, since it implies an attempt to ease tension; or as negative, since it could be viewed as a sign of weakness. Another example is *unantastbar*, which again can be interpreted positively or negatively.

The controversial words generally denote abstract notions, or have established metaphorical senses. In the authors' view, their polarity is heavily context-dependent, and a one-dimensional score is not sufficient to model their contribution to sentiment.

In fact, most of these words have been assigned very small polarity values in the combined lexicon, which reflects the conflicting evidence present in the various sources. Of the 22 items which differ in C&K, the one with the largest value in the combined lexicon is *dominieren*, which has been assigned a fairly negative combined score, but was rated positive (0.5) in C&K.

5.7 Distribution

We are making SentiMerge freely available for download. The current version is available at <https://github.com/guyemerson/SentiMerge>

5.8 Future Work

To align the disparate sources, a simple linear rescaling was used. However, in principle any monotonic function could be considered. A more general function that would still be tractable could be $u_i \mapsto \lambda u_i^\alpha$.

Furthermore, the probabilistic model described in section 5.5.1 makes several simplifying assumptions, which could be weakened or modified. For instance, we have assumed a normal distribution, with zero mean, both for the prior distribution and for the error terms. The data is not perfectly modelled by a normal distribution, since there are very clear bounds on the polarity scores, and some of the data takes discrete values. Indeed, we can see in figure 5.1 that the data is not normally distributed. An alternative choice of distribution might yield better results.

More generally, our method can be applied to any context where there are multiple resources to be merged, as long as there is some real-valued property to be aligned.

Chapter 6

Evaluation report for the new domain-specific language data and tools

6.1 Introduction

The TrendMiner consortium has been extended in its last year with four new partners (see the second page of this deliverable for the complete list of partners). The new partners are the Institute of Computer Science at the Polish Academy of Sciences (IPIAN), the Research Institute for Linguistics of the Hungarian Academy of Sciences (RILMTA), Daedalus S.A. (DAEDALUS) and Universidad Carlos III de Madrid (UC3M). Those partners have developed resources and linguistic processors for new or extended use cases of TrendMiner. The domains covered are health, finance, psychology and politics. For the financial and political domains, links and collaborations have been established to work done in the context of WP6 and WP7. Concerning the health use case, a web service has been developed that shows how drugs, diseases and adverse effects of drugs are mentioned in social media, including blogs and social networks in Spanish. For each extended use case, adaptation and extension of the TrendMiner ontologies (described in D2.1.2: Knowledge and Provenance Modelling and Stream Modelling) Multilingual resources and evaluation of knowledge modelling) developed in WP2 have been performed. Most of this work is described in D2.4: Integration of lexical and terminological data in the TrendMiner platform.

6.2 Links to evaluation reports in D10.1

As Deliverable 10.1 "Newly generated domain-specific language data and tools" is giving a detailed description of the work performed by the new partners of TrendMiner, including evaluation reports, we prefer to point to the relevant sections of this D10.1 for a description of the evaluation studies conducted by the new partners.

Concerning the health use case, in Spanish language, an evaluation report is available in section 2.4 of D10.1.

Evaluation reports for the extended political use case are available for the Polish language under section 4.4 and for the Hungarian language under section 5.5

Chapter 7

Conclusion

We have presented in this deliverable evaluation studies related to multilingual information extraction, to a-priori lexical knowledge for sentiment analysis and for analytic tools developed by the new partners of the consortium. Related work to the extended knowledge sources (ontologies) for TrendMiner is described in D2.4.

Bibliography

- [ACG⁺12] Enrique Amigó, Adolfo Corujo, Julio Gonzalo, Edgar Meij, and Maarten de Rijke. Overview of RepLab 2012: Evaluating Online Reputation Management Systems. In *CLEF 2012 Labs and Workshop Notebook Papers*, 2012.
- [AGB⁺12] Niraj Aswani, Mark Greenwood, Kalina Bontcheva, Leon Derczynski, Julian Moreno Schneider, Hans-Ulrich Krieger, and Thierry Declerck. Multilingual, ontology-based information extraction from stream media - v1. Technical Report D2.2.1, TrendMiner Project Deliverable, 2012.
- [AGBP13] Niraj Aswani, Genevieve Gorrell, Kalina Bontcheva, and Johann Petrak. Multilingual, ontology-based information extraction from stream media - v2. Technical Report D2.2.2, TrendMiner Project Deliverable, 2013.
- [App99] Douglas E. Appelt. An Introduction to Information Extraction. *Artificial Intelligence Communications*, 12(3):161–172, 1999.
- [BLK⁺09] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. DBpedia – a crystallization point for the web of data. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 7:154–165, 2009.
- [Car97] C. Cardie. Empirical Methods in Information Extraction. *AI Magazine*, 18(4), 1997.
- [CGK⁺12] Simon Clematide, Stefan Gindl, Manfred Klenner, Stefanos Petrakis, Robert Remus, Josef Ruppenhofer, Ulli Waltinger, and Michael Wiegand. MLSA – a multi-layered reference corpus for German sentiment analysis. pages 3551–3556. European Language Resources Association (ELRA), 2012.
- [CK10] Simon Clematide and Manfred Klenner. Evaluation and extension of a polarity lexicon for German. In *Proceedings of the First Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, page 7, 2010.
- [DJHM13] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems, I-SEMANTICS '13*, pages 121–124, New York, NY, USA, 2013. ACM.

- [DK14] Thierry Declerck and Hans-Ulrich Krieger. TMO – the federated ontology of the TrendMiner project. In *Proceedings of the 9th International Language Resources and Evaluation Conference (LREC 2014)*, 2014.
- [DMAB13a] L. Derczynski, D. Maynard, N. Aswani, and K. Bontcheva. Microblog-Genre Noise and Impact on Semantic Annotation Accuracy. In *Proceedings of the ACM Conference on Hypertext and Social Media (HT’13)*, 2013.
- [DMAB13b] L. Derczynski, D. Maynard, N. Aswani, and K. Bontcheva. Microblog-Genre Noise and Impact on Semantic Annotation Accuracy. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*. ACM, 2013.
- [DYJ13] L. Derczynski, B. Yang, and C.S. Jensen. Towards Context-Aware Search and Analysis on Social Media Data. In *Proceedings of the 16th Conference on Extending Database Technology*. ACM, 2013.
- [EKG13] Judith Eckle-Kohler and Iryna Gurevych. The practitioner’s cookbook for linked lexical resources. 2013.
- [FS12] Paolo Ferragina and Ugo Scaiella. Fast and accurate annotation of short texts with wikipedia pages. *IEEE Software*, 29(1):70–75, 2012.
- [GS96] Ralph Grishman and Beth Sundheim. Message understanding conference - 6: A brief history. In *Proceedings of the 16th Conference on Computational linguistics (COLING’96)*, Copenhagen, Denmark, 1996.
- [HAW14] Johannes Hoffart, Yasemin Altun, and Gerhard Weikum. Discovering emerging entities with ambiguous names. In *Proceedings of the 23rd International Conference on World Wide Web, WWW ’14*, pages 385–396, Republic and Canton of Geneva, Switzerland, 2014. International World Wide Web Conferences Steering Committee.
- [HB11] B. Han and T. Baldwin. Lexical Normalisation of Short Text Messages: Makn Sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT’11)*, 2011.
- [HC97] Lynette Hirschmann and Nancy Chinchor. MUC-7 coreference task definition. In *Proceedings of MUC-7*, 1997.
- [HF97] Birgit Hamp and Helmut Feldweg. GermaNet – a lexical-semantic net for German. In *Proceedings of the ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pages 9–15. Association for Computational Linguistics, 1997.
- [HYB⁺11] J. Hoffart, M. A. Yosef, I. Bordino, H. Furstenuau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust disambiguation of named entities in text. In *Conference on Empirical Methods in Natural Language Processing, Edinburgh, Scotland, 2011*, pages 782–792. Morgan Kaufmann, California, 2011.

- [JGD⁺10] H. Ji, R. Grishman, H. T. Dang, K. Griffitt, and J. Ellis. Overview of the tac 2010 knowledge base population track. In *Proceedings of the Third Text Analysis Conference (TAC 2010)*, 2010.
- [JSFT07] A. Java, X.D. Song, T. Finin, and B. Tseng. Why We Twitter: Understanding Microblogging. Usage and Communities. In *Proceedings of the Workshop on Web mining and social network analysis*, 2007.
- [LZW⁺12] X. Liu, M. Zhou, F. Wei, Z. Fu, and X. Zhou. Joint inference of named entity recognition and normalization for tweets. In *Proceedings of the Association for Computational Linguistics*, pages 526–535, 2012.
- [MBR12] D. Maynard, K. Bontcheva, and D. Rout. Challenges in developing opinion mining tools for social media. In *Proceedings of the @NLP can u tag #usergeneratedcontent?! workshop at LREC’12*, 2012.
- [MW08a] D. Milne and I. H. Witten. Learning to link with Wikipedia. In *Proc. of the 17th Conf. on Information and Knowledge Management (CIKM)*, pages 509–518, 2008.
- [MW08b] David Milne and Ian H. Witten. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *In Proceedings of AAAI 2008*, 2008.
- [MWdR12] E. Meij, W. Weerkamp, and M. de Rijke. Adding semantics to microblog posts. In *Proc. of the Fifth Int. Conf. on Web Search and Data Mining (WSDM)*, 2012.
- [RBH13] D. Rout, K. Bontcheva, and M. Hepple. Reliably evaluating summaries of twitter timelines. In *Proceedings of the AAAI Workshop on Analyzing Microtext*, 2013.
- [RCME11] A. Ritter, S. Clark, Mausam, and O. Etzioni. Named Entity Recognition in Tweets: An Experimental Study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP’11)*, 2011.
- [RMD13] D. Rao, P. McNamee, and M. Dredze. Entity linking: Finding extracted entities in a knowledge base. In *Multi-source, Multi-lingual Inf. Extraction and Summarization*. Springer, 2013.
- [RQH10] Robert Remus, Uwe Quasthoff, and Gerhard Heyer. SentiWS – a publicly available German-language resource for sentiment analysis. In *Proceedings of the 7th International Language Resources and Evaluation Conference (LREC 2010)*, 2010.
- [RR09] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155. Association for Computational Linguistics, 2009.
- [RSD⁺13] M. Rowe, M. Stankovic, A.S. Dadzie, B.P. Nunes, and A.E. Cano. Making sense of microposts (#msm2013): Big things come in small packages. In *Proceedings of the WWW Conference - Workshops*, 2013.

- [RUH⁺14] Michael Rder, Ricardo Usbeck, Sebastian Hellmann, Daniel Gerber, and Andreas Both. N3 - a collection of datasets for named entity recognition and disambiguation in the nlp interchange format. In *The 9th edition of the Language Resources and Evaluation Conference*, 26-31 May, Reykjavik, Iceland, 2014.
- [RYZ⁺10] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *The Journal of Machine Learning Research*, 11:1297–1322, 2010.
- [SDS66] Philip J Stone, Dexter C Dunphy, and Marshall S Smith. The general inquirer: A computer approach to content analysis. 1966.
- [SG02] Joseph L Schafer and John W Graham. Missing data: our view of the state of the art. *Psychological methods*, 7(2):147, 2002.
- [SOSK04] Kiril Simov, Petya Osenova, Alexander Simov, and Milen Kouylekov. Design and implementation of the bulgarian hpsg-based treebank. *Research on Language and Computation*, 2(4):495–522, 2004.
- [TIO05] Hiroya Takamura, Takashi Inui, and Manabu Okumura. Extracting semantic orientations of words using spin model. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2005.
- [vERT13] M. van Erp, G. Rizzo, and R. Troncy. Learning with the Web: Spotting Named Entities on the intersection of NERD and Machine Learning. In *Proceedings of the 3rd Workshop on Making Sense of Microposts (#MSM2013)*, 2013.
- [Wal10a] Ulli Waltinger. GermanPolarityClues: A lexical resource for German sentiment analysis. In *Proceedings of the 7th International Language Resources and Evaluation Conference (LREC 2010)*, 2010.
- [Wal10b] Ulli Waltinger. Sentiment analysis reloaded - a comparative study on sentiment polarity identification combining machine learning and subjectivity features. In *Proceedings of the 6th International Conference on Web Information Systems and Technologies (WEBIST 2010)*. INSTICC Press, 2010.
- [WWH05] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics, 2005.