EACL 2014

14th Conference of the European Chapter of the Association for Computational Linguistics

Proceedings of the Student Research Workshop

April 26-30, 2014
Gothenburg, Sweden

GOLD SPONSORS

Google™

معهد قطر لبحوث الحوسبة
Qatar Computing Research Institute
Member of Qatar Foundation عضو في مؤسسة قطر

SILVER SPONSOR

FINDWISE
SEARCH DRIVEN SOLUTIONS

BRONZE SPONSORS

facebook

Yandex

SUPPORTERS

W
UNIVERSITY *of*
WASHINGTON

UNIVERSITY OF
GOTHENBURG

MASTER'S PROGRAMME IN
LANGUAGE
TECHNOLOGY

Talkamatic
FREE DIALOGUE

EXHIBITORS

M&C
MORGAN & CLAYPOOL
PUBLISHERS

Springer

OTHER SPONSORS

UNIVERSITY OF
GOTHENBURG

City of
Gothenburg

SAS OFFICIAL
AIRLINE
A STAR ALLIANCE MEMBER

HOSTS

CLT
Centre for Language Technology
Gothenburg, Sweden

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-937284-80-0

Introduction

Welcome to the EACL 2014 Student Research Workshop.

This workshop continues the tradition of providing a forum for student researchers and builds on the success of the previous workshops held in Bergen (1999), Toulouse (2001), Budapest (2003), Trento (2006), Athens (2009), and Avignon (2012). It is an excellent venue for student researchers investigating topics in computational linguistics and natural language processing to present and discuss their research, to meet potential advisors and leading world experts in their research fields, as well as to receive feedback from the international research community.

This year we have introduced two different types of submissions: research papers and thesis proposals. Thesis proposals are intended for advanced students who have decided on a thesis topic and wish to get feedback on their proposal and broader ideas for their continuing work, while research papers can describe completed work or work in progress with preliminary results. All accepted research papers are presented as talks in two separate sessions allocated for the workshop during the main EACL 2014 conference, while all accepted thesis proposals are presented as posters during the main EACL 2014 poster session.

On behalf of the entire Program Committee, we are delighted to present the proceedings of the Student Research Workshop. We received 13 thesis proposals and 29 research papers. We accepted 5 thesis proposals and 8 research papers leading to an acceptance rate of 38% for thesis proposals and 28% for research papers. The overall quality of the submissions was high, and we thank our program committee for their dedicated and thorough work and excellent feedback.

We also thank our faculty advisor Sebastian Padó for his suggestions, feedback and his extremely quick responding to all our questions. We also thank the EACL 2014 organizing committee, especially Shuly Wintner, Stefan Riezler, Sharon Goldwater, Nina Tahmasebi, Gosse Bouma and Yannick Parmentier, for providing us advice and assistance in planning and organizing this workshop. We also want to thank the EACL for providing financial support for students who would otherwise be unable to attend the workshop and the conference.

We truly hope you will enjoy the Student Research Workshop in Gothenburg!

Desmond Elliott, University of Edinburgh
Konstantina Garoufi, University of Potsdam
Douwe Kiela, University of Cambridge
Ivan Vulić, KU Leuven

EACL 2014 Student Research Workshop Co-Chairs

Student Chairs:

Desmond Elliott, *University of Edinburgh*
Konstantina Garoufi, *University of Potsdam*
Douwe Kiela, *University of Cambridge*
Ivan Vulić, *KU Leuven*

Faculty Advisor:

Sebastian Padó, *University of Stuttgart*

Program Committee:

Marianna Apidianaki, *LIMSI-CNRS*
Borja Balle, *McGill University*
Marco Baroni, *University of Trento*
Timo Baumann, *University of Hamburg*
Lee Becker, *Hapara*
Steven Bethard, *University of Alabama at Birmingham*
Chris Biemann, *TU Darmstadt*
Hendrik Buschmeier, *University of Bielefeld*
Marcela Charfuelan, *DFKI*
Christian Chiarcos, *University of Potsdam*
Laurence Danlos, *Université Paris 7*
Jan De Belder, *KU Leuven*
Vladimir Eidelman, *University of Maryland*
Jacob Eistenstein, *Georgia Tech*
Antske Fokkens, *University of Amsterdam*
Goran Glavaš, *University of Zagreb*
João Graça, *Inesc-ID*
Weiwei Guo, *Columbia University*
Dilek Hakkani-Tür, *Microsoft Research*
Bo Han, *University of Melbourne*
Katja Hofmann, *Microsoft Research*
Ann Irvine, *Johns Hopkins University*
David Jurgens, *Sapienza University*
Anna Kazantseva, *University of Ottawa*
Philipp Koehn, *University of Edinburgh*
Oleksandr Kolomiyets, *KU Leuven*
Sebastian Krause, *DFKI*
Vasileios Lampos, *University College London*
Els Lefever, *Ghent University*
Pierre Lison, *University of Oslo*
Elijah Mayfield, *Carnegie Mellon University*
Roser Morante, *University of Antwerp*
Preslav Nakov, *Qatar Computing Research Institute*
Dong Nguyen, *University of Twente*
Joakim Nivre, *Uppsala University*
Gabriella Pasi, *University of Milano Bicocca*

Alexandre Passos, *UMass Amherst*
Andreas Peldszus, *University of Potsdam*
Tamara Polajnar, *University of Cambridge*
Ariadna Quattoni, *UPC*
Sravana Reddy, *Dartmouth College*
Michaela Regneri, *Saarland University*
Christian Scheible, *University of Stuttgart*
Amanda Stent, *Yahoo! Labs*
Kristina Striegnitz, *Union College*
Anders Søgaard, *University of Copenhagen*
Joel Tetreault, *Yahoo! Labs*
Oscar Täckström, *Google*
Tim Van de Cruys, *IRIT*
Svitlana Volkova, *Johns Hopkins University*
Mengqiu Wang, *Stanford University*
Rui Wang, *DFKI*
Jason Williams, *Microsoft Research*
Alessandra Zarcone, *University of Stuttgart*
Jan Šnajder, *University of Zagreb*

Table of Contents

Student Research Workshop Program

Monday, April 28, 2014

15:15–18:30 **Posters (Thesis Proposals)**

Literature-Based Discovery for Oceanographic Climate Science
Elias Aamot

Unsupervised Relation Extraction of In-Domain Data from Focused Crawls
Steffen Remus

Enhancing Medical Named Entity Recognition with Features Derived from Unsupervised Methods
Maria Skeppstedt

Now We Stronger than Ever: African-American English Syntax in Twitter
Ian Stewart

Expanding the Range of Automatic Emotion Detection in Microblogging Text
Jasy Suet Yan Liew

Tuesday, April 29, 2014

12:30–14:00 SRW Lunch

16:30–18:10 **Parallel Session I (Research Papers I)**

16:30–16:55 *Resolving Coreferent and Associative Noun Phrases in Scientific Text*
Ina Roesiger and Simone Teufel

16:55–17:20 *Modelling Irony in Twitter*
Francesco Barbieri and Horacio Saggion

17:20–17:45 *Multi-class Animacy Classification with Semantic Features*
Johannes Bjerva

17:45–18:10 *Using Minimal Recursion Semantics for Entailment Recognition*
Elisabeth Lien

16:30–18:10 **Parallel Session II (Research Papers II)**

16:30–16:55 *A Graph-Based Approach to String Regeneration*
Matic Horvat and William Byrne

16:55–17:20 *Complexity of Word Collocation Networks: A Preliminary Structural Analysis*
Shibamouli Lahiri

Tuesday, April 29, 2014 (continued)

16:30–18:10 **Parallel Session II (Research Papers II) (continued)**

17:20–17:45 *Automatic Creation of Arabic Named Entity Annotated Corpus Using Wikipedia*
Maha Althobaiti, Udo Kruschwitz and Massimo Poesio

17:45–18:10 *Generating Artificial Errors for Grammatical Error Correction*
Mariano Felice and Zheng Yuan

Literature-based discovery for Oceanographic climate science

Elias Aamot
Department of Informatics and Computer Science
Norwegian University of Science and Technology
Trondheim, Norway
eliasaa@stud.ntnu.no

Abstract

This paper presents an overview of the field of literature-based discovery, as originally applied in biomedicine. Furthermore it identifies some of the challenges to employing the results of the field in a new domain, namely oceanographic climate science, and elaborates on some of the research that needs to be conducted to overcome these challenges.

1 Introduction

The increase in growth rate of the scientific literature over the past decades has forced researchers to become increasingly specialized in order to keep up with the state of the art. This inevitably leads to the fragmentation of science as researchers from different (sub-)disciplines rarely have time to read each other's papers. Swanson (1986) claimed that this fragmentation of science can lead to *undiscovered public knowledge*: Conclusions that can be made from existing literature, but have never been made because the knowledge fragments have been discovered in separate (sub-)disciplines. Adopting the terminology of Swanson (1991), a literature can be informally defined as a collection of papers with a significant amount of cross-citation related to a single topic. Two literatures are *complementary* if they contain knowledge fragments which can be combined to form new knowledge, and *disjoint* if they have no articles in common, and exhibit little or no cross-citation. The implicit hypothesis is that such *complementary but disjoint* (CBD) literatures are common, giving rise to significant amounts of undiscovered public knowledge. The field of *Literature-based Discovery* (LBD)[1] focuses on the development and application of computational tools to discover undiscovered public knowledge in scientific literature.

Most work in LBD has been conducted in subfields of the biomedical literature, frequently employing knowledge resources specific to that domain. This paper will present an overview of some of the research in LBD, and discuss some of the challenges in reproducing the results made in the LBD field in a different domain, namely oceanographic climate science. The structure of this paper is as follows: Section 2 will give an overview of the LBD field, section 3 will discuss differences between the biomedical domain and that of oceanographic climate science, and section 4 will discuss directions for research that will be conducted in order to adapt LBD methods to the oceanographic climate science domain.

2 Literature-based discovery

Swanson (1986) observed that if a literature L_1 asserted $a \rightarrow b$, and a disjoint literature L_2 asserted $b \rightarrow c$, then the concept denoted by b could function as a bridge between L_1 and L_2, leading to the discovery of the hypothesis $a \rightarrow c$[2]. One example given by Swanson showed that fish oils reduced blood viscosity ($fish\ oil \rightarrow blood\ viscosity$), and that patients of Raynaud's disease tend to exhibit high blood viscosity ($blood\ viscosity \rightarrow Raynaud$). These two facts led to the hypothesis that fish oils can be used in the treatment of Raynaud's disease ($fish\ oil \rightarrow Raynaud$) when combined. This hypothesis was subsequently confirmed experimentally (Digiacomo et al., 1989). Although the inference steps are not logically sound, the procedure is able to produce interesting results. The general approach of bridging dis-

[1] Also called *Literature-based knowledge discovery* (LBKD).

[2] A note on terminology: In the LBD literature, capital letters are normally used for the A, B and C concepts. In this paper, minuscules will be used to represent individual concepts, while capital letters represent sets.

Also, some authors use A to denote the the goal concept, and C for the starting concept. This paper follows the most commonly used terminology, in which a always denotes the starting concept, and c denotes the goal concept.

Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 1–10,
Gothenburg, Sweden, April 26-30 2014. ©2014 Association for Computational Linguistics

joint literatures by means of intermediary terms has been dubbed *Swanson linking*, and is also referred to as the *ABC model*.

Swanson and Smalheiser (1997) explain that the discovery of the ABC structure and the fish oil-Raynaud's disease connection happened accidentally. This discovery led Swanson to conduct literature searches aided by existing information retrieval tools to search for more undiscovered public knowledge using the ABC model, resulting in the discovery of eleven connections between migraine and magnesium (Swanson, 1988). As the discovery process was extremely time consuming, requiring the researcher to read hundreds of papers, Swanson later developed a computational tool, Arrowsmith, to streamline the discovery process.

There are two modes of discovery in the ABC model: *Open discovery* and *closed discovery*. In open discovery, the researcher only knows the starting concept a, and is interested in uncovering undiscovered public knowledge related to a. A researcher who looks for consequences of ocean acidification might conduct an open discovery search with $a = ocean\ acidification$. In closed-discovery, the researcher knows both the starting concept a and the goal concept c, and is interested in finding concepts B that prove an explanation of the relationship between the two terms. A researcher who hypothesizes that ocean acidification might cause a reduction in phytoplankton population and tries to discover the causality chain might conduct a closed discovery search with $a = ocean\ acidification, c = phytoplankton\ population$.

This section will present an overview of the state-of-the-art of the LBD field. As this paper discusses the adaptation of LBD to new domains, approaches will be grouped into of three groups according to their dependence on domain specific tools and resources, because reliance on these is likely to hinder cross-domain adaptation[3].

2.1 Group 1: Domain-independent approaches

In the general Swanson linking paradigm, open discovery is conducted by extracting all relations $a \rightarrow b_i$ from the literature of a, written $L(a)$. For every b_i, all relations $b_i \rightarrow c_j$ are then extracted from $L(b_i)$. The set of all $a \rightarrow b_i \rightarrow c_j$ relations, dubbed *discovery candidates* is then are presented to the user as potential discoveries, sorted according to some ranking metric.

In most LBD approaches $L(x)$ is defined as the set of documents returned when searching for x in a literature database. The literature database most commonly used in LBD is Pubmed/Medline[4], maintained by the US National Library of Medicine. The original Arrowsmith system considered only paper titles, as Swanson considered these to hold the most compact knowledge, but it has become the standard approach in LBD to use abstracts and possibly index terms in addition to the titles. The motivation for this is that abstracts and index terms contain more knowledge than only titles.

Somewhat surprisingly, few LBD systems use full paper texts. Schuemie et al. (2004) show that 30-40% of all information contained in a section is new to that section, meaning that significant amounts of knowledge is lost when only looking at abstracts and index terms of a paper. The need for full text data is also pointed out by Cameron et al. (2013). The reason for not using full text seems to be that paper abstracts and index terms are available in xml format through the Pubmed API, while full paper texts require accessing rights and are normally stored as pdf.

In co-occurrence based systems, a relation $x \rightarrow y$ is postulated if x and y exhibit a high degree of co-occurrence in $L(x)$, either in terms of absolute frequency of co-occurrence, or in terms of statistical unlikelihood given the statistical promiscuity of the two concepts. While a few systems use the sentence as the domain for counting co-occurrences, most systems count co-occurrences across entire abstracts.

To present the user with only potential new discoveries, most LBD systems remove from C all terms that are already known to be in a relation with a. In co-occurrence based methods, this is done by removing any (a, c) pairs that exhibit higher degrees if co-occurrence than a predefined threshold (normally 1 co-occurrence) in $L(a)$.

2.1.1 Arrowsmith

The original Arrowsmith system works as follows (Swanson and Smalheiser, 1997): $L(a)$ is fetched

[3] Some of the papers are presented as domain independent, even though they employ domain specific resources, because their main research contributions can be adapted in a domain-independent manner.

[4] http://www.ncbi.nlm.nih.gov/pubmed/

by conducting a Medline search to retrieve the titles of papers containing a in the title. The set of potential B concepts is extracted as the list of unique words in $L(a)$, after a stop list of approximately 5000 words has been applied. The B-term set is further pruned by removing all the words that have lesser relative frequency in $L(a)$ than in Medline. The potential B terms are subsequently presented to the user, who can then remove words that are thought to be unsuitable. For each $b_i \in B$, $L(b_i)$ is retrieved and a set C_i is generated, subject to the same stopword and frequency restrictions as before. The terms in the union of the C_i sets are then ranked according to the number of b-terms that connect them to the a-term.

2.1.2 Information retrieval-based methods

Gordon and Lindsay (1996) (Lindsay and Gordon, 1999) developed a system in parallel, which differed from Arrowsmith in several ways: Firstly, while Arrowsmith was word-based, their system used n-grams as the unit of analysis. A stop list was applied by removing all n-grams that contained any stop word occurrence. Secondly, their system used entire Medline records, comprising of keywords, abstracts and titles, whereas Arrowsmith only used paper titles. Thirdly, their system employed information retrieval metrics such as *tf*idf* to find b-terms among the generated candidates, whereas Arrowsmith was based on relative frequencies.

The lexical statistical approach is so generic that it lends itself directly to application in different domains. In a later paper, Gordon et al. (2001) employ this approach to conduct LBD searches directly on the World Wide Web, searching for application areas for genetic algorithms. It should however be noted that the goal of this experiment was not LBD in the sense of uncovering undiscovered public knowledge, instead focusing in discovering something that might be "publicly known" but novel to the user.

2.1.3 Ranking metrics

Wren et al. (2004) pointed out that the structure of concept co-occurrence relationships is such that most concepts are connected to any other concept within few steps. This *small world phenomenon* implies that research focus should be shifted away from retrieving discovery candidates to ranking them, because a significant portion of the concept space will be retrieved even within two co-occurrence relation steps. The paper proposes ranking implicit relationships by comparing the number of observed indirect connections between a and c to the number of expected connections in a random network model, given the relative promiscuity of the intermediary terms.

In another paper, Wren (2004) emphasizes the importance of using a statistically sound method of ranking relationship strengths, such as "chi-square tests, log-likelihood ratios, z-scores or t-scores", because co-occurrence based measures bias towards more general, and thus less interesting relationships. The paper further proposes an extension to the mutual information measure (MIM) as a ranking measure.

2.1.4 Latent semantic indexing

Gordon and Dumais (1998) propose exploiting the ability of certain vector-based semantic models such as Latent semantic indexing (LSI) to discover implicit relationships between terms for LBD. They first train the semantic model on $L(a)$, and let the user choose as b one of the terms most similar to a. A new semantic model is built from $L(b)$, and discovery candidates are ranked according to their similarity to a in the $L(b)$-model. Their experiments showed that the resulting b- and c-term candidate lists closely resemble the lists produced by the information retrieval inspired lexical statistics.

In another experiment they built a semantic model from a random sample of all of Medline, and looked directly for c-terms in the semantic model by considering the terms most similar to a. This "zoomed-out" approach produced different results than the previous Swanson linking inspired approach, which the authors claimed meant that the two methods are complementary and could therefore be used in parallel, but no in-depth evaluation was conducted on the quality of the results.

2.1.5 Evaluation efforts

LBD has a tradition for questionable evaluation effort. The original discoveries in LBD were made manually by Swanson, and most computational systems are evaluated solely according to their ability to replicate one or more of Swanson's discoveries. This is problematic for several reasons: First of all, Swanson's discoveries were never intended as a gold standard, and being able to accomplish a single task that is known in advance does not mean that the results are generalizable.

Secondly, there is no quantitative basis for comparing different approaches or metrics.

Yetisgen-Yildiz and Pratt (2009) conducted the first systematic quantitative evaluation of discovery candidate ranking metrics and relation ranking/generation techniques. They partitioned Medline into two parts, according to a cut-off date. LBD was conducted on the pre-cut-off set, and the post-cut-off set was used as a gold standard to compute precision and recall. In the post-cut-off set, a connection was considered to exist if two terms co-occurred in any document. The ranking metrics that were evaluated were Linking term count (LTC), that is the number of b-terms connecting a and c, Average minimum weight (AMW), that is the average weight of the $a \rightarrow b \rightarrow c$ connections, and Literature cohesiveness (COH), a measure developed by Swanson but not widely adopted. Experiments showed that LTC gave better precision at all levels of recall. The relation generation techniques that were considered were association rules, tf-idf, z-score and MIM. The experiment showed that association rules give the best precision score (8.8%) but the worst recall score (53.76%), while tf-idf gave the best recall (88.0%) but a rather low precision (2.29%).

While the evaluation effort was an important contribution to the LBD field, more quantitative evaluation is required. First of all, all candidate ranking/generation techniques and ranking metrics were tested with only one value of the parameters (for instance the cut-off score for tf-idf, and the cut-off probability for z-score). Comparing the performance of different settings for the parameters would yield a better understanding of each of the metrics, and could lead to results completely different than those reported. Secondly, only a small subset of possible relation generation/ranking techniques and discovery candidate ranking metrics were tested. For example, no relation extraction-based methods (see section 2.3) were included in the evaluation.

The evaluation methodology can be critiqued in several ways. Firstly, building the gold standard from the post-cut-off set is problematic for several reasons: A co-occurrence can exist in the post-cut-off set without necessarily corresponding to a new discovery. Also, as pointed out in Kostoff (2007), it is very difficult to verify that a discovery has not been made before the cut-off date. Another problem is that the post-cut-off set only contains discoveries that have been made in the present, all future discoveries are therefore excluded from the gold standard. Secondly, it is not obvious that quantitative measures reflect the usefulness of the LBD system: When at all is said and done, the usefulness of a LBD system equates to its ability to support user in discovering knowledge.

2.2 Group 2: Concept-based approaches

Several researchers advocate using domain specific concepts taken from an ontology or controlled vocabularies instead of n-gram tokens. Using concepts provides three benefits over n-gram models: Firstly, synonyms and spelling variants are mapped to the same semantic concept. Secondly, using concepts allows for ranking and filtering according to semantic categories. Finally, it becomes easier to constrain the search space by removing spurious or irrelevant n-grams at an early stage, as they don't map to any concept in the domain. On the other hand, concept extraction from raw text is a non-trivial operation.

In LBD concept extraction is conducted in one of two ways: One option is to use NLP tools designed for entity recognition. The most commonly used in the biomedical domain is *MetaMap* (Aronson and Lang, 2010), which extracts concepts from the Unified Medical Language System (UMLS) meta-thesaurus[5]. The other option is to use Medical Subject Headings (MeSH)[6]. MeSH is a controlled vocabulary for indexing biomedical papers, with which all Medline papers have been manually tagged. MeSH keywords can be queried directly from the Medline API. Both MeSH and UMLS terms are organized hierarchically according to semantic categories.

2.2.1 DAD

In their system, DAD (Disease-Adverse reaction-Drug), Weeber et al. (2001) use MetaMap. They showed in an experiment that the number of concepts extracted is significantly lower than the number of n-grams, even after stop lists are applied (8,362 n-grams vs. 5,998 concepts). DAD also allows the user to specify which semantic categories to consider, by for instance only allowing concepts of the type *pharmacological substance* as c concepts, reducing the number of search paths significantly.

[5] http://www.nlm.nih.gov/research/umls/

[6] http://www.nlm.nih.gov/mesh/

Their approach was able to replicate both Swanson's *Raynaud's-fish oil* and *migraine-magnesium* discoveries, but it was discovered that MetaMap maps both *mg* (milligram) and *Mg* (magnesium) to the concept *magnesium*, giving optimistic results for the migraine-magnesium experiment. This is but one example showing that one of the problems with employing NLP tools in an LBD system is that system performance becomes closely tied to the performance of the tools it employs.

2.2.2 LitLinker

Pratt and Yetisgen-Yildiz (2003) developed a system, LitLinker, which originally also used MetaMap, but they later found it too computationally expensive for practical use (Yetisgen-Yildiz and Pratt, 2006). MeSH terms are therefore employed instead.

In a preprocessing step, LitLinker calculates the co-occurrence patterns of every MeSH term across the literatures of every other MeSh term. For every MeSH term, the mean and standard deviation of co-occurrence counts across the literatures is calculated. In the discovery process, a term is considered to be related to another term if their co-occurrence is higher than statistically expected, based on its z-score.

Yetisgen-Yildiz and Pratt identified three classes of uninteresting links and terms that should be pruned automatically by system: (1) too broad terms (giving the examples *medicine*, *disease* and *human*), (2) too closely related terms (giving the example *migraine* and *headache*), and (3) semantically nonsensical connections. The first class is handled by removing any concept if it is strictly more specific in the MeSH ontology hierarchy than any included term. The second class is handled by pruning all links between terms that are closely related (grandparents, parents, siblings and children) in the ontology. The third class is handled by letting the user specify which semantic classes of concepts are allowed to link.

2.2.3 Bitola

Hristovski et al. (2001) originally developed a system called Bitola[7] that discovered *association rules* between MeSH terms. Association rules mining is a common data mining method for discovering relations between variables in a database. Association rules are traditionally used for market basket analysis, in which rules of the type $\{pizza, steak\} \rightarrow \{coca\ cola\}$ are inferred, stating that if somebody buys pizza and steak, he/she is likely to buy coca cola as well. In Bitola's discovery step, basic associations are first mined from the co-occurrence patterns of MeSH terms. Subsequently, indirect associations $a \rightarrow c$ are inferred by combining association rules on the form $a \rightarrow b_i$ and $b_i \rightarrow c$, and ranked according to the sum of strengths of the connecting association rules.

2.3 Group 3: Relation extraction-based approaches

Hristovski et al. (2006) point out two problems with the co-occurrence based LBD systems: Firstly, no explicit explanation of the relation between the a and c terms is given. Secondly, a large number of spurious relations are discovered, as demonstrated by the low precision values witnessed during system evaluation. Both aspects increase the time needed to examine the output of the system by the human user. They suggest that employing natural language processing (NLP) techniques to extract explicit relations from the papers can improve performance on both points.

The biomedical information extraction tool most commonly used in LBD is *SemRep* (Rindflesch and Fiszman, 2003), which uses linguistically motived rules on top of the ouput from MetaMap and the Xerox POS Tagger to extract knowledge in the form of $< subject, predicate, object >$ relation triplets. Although the knowledge expressed in natural language is more complex than what can be represented in simple relation triplets, SemRep is able to provide a better approximation to the knowledge content of scientific papers than do co-occurrence based methods.

While most LBD research employs the same NLP tool, systems differ as to how the extracted relations are represented and how reasoning is conducted in the relation space. Some researchers closely follow the Swanson linking paradigm, and use relation extraction based method instead of or in addition to co-occurrence based methods for candidate generation and ranking. Other researchers take an approach motivated by Wren's observation that a small-world property holds in the network of concept relations in literature. As significant portions of the concept-relation space will have to be explored in a two-step search anyway, it might be better to extract all relations from

[7] http://ibmi3.mf.uni-lj.si/bitola/

the entire literature collection or from a random sample thereof, and rather focus on valid and efficient reasoning within the entire concept-relation space.

Smalheiser (2012) critiques the usage of relation extraction in LBD and claims that while reasoning over explicit relations may lead to so-called *incremental discoveries*, that is, discoveries that lie close to the existing knowledge and therefore are less interesting, they are not able to lead to any *radical discoveries*, that is discoveries that seem unlikely at time of discovery. He also claims that human discoveries, both incremental and radical, tend to be on a higher level, using analogies and abstract similarities rather than explicit relations, and that the benefit from using relation extraction therefore is minimal[8].

2.3.1 Augmented Bitola

In two papers, Hristovski et al. (2006; 2008) experiment with augmenting the Bitola system by using relation extraction tools. In addition to SemRep, they also use another tool, *BioMedLee*, because each of the tools exhibits better performance than the other on certain types of relations.

To guide search through the concept-relation space, they introduce the notion of a *discovery pattern*. A discovery pattern is a set of concept types and relations between them that could imply an interesting relationship in the domain. One discovery pattern, *maybe_treats* can informally be stated as: If a disease leads to a biological change, and a drug leads to the opposite change, then the drug may be able to treat the disease.

The integration between Bitola and the NLP components presented in the system is rather crude; for a given query term, Bitola outputs a set of related terms and the set of papers connecting each related term to the query term. The connecting paper must then be manually input into the NLP components to extract the relation between the query term and any related term. Following a discovery pattern requires extracting relations between several concepts until a chain of the correct relations has been found. The possibility to integrate Bitola and the NLP tools more tightly has been raised as possible future work, but it has been noted a concern that the computational load increases as the NLP component becomes less constrained by the co-occurrence based components.

2.3.2 Graph-based reasoning

The extracted relations can be represented as a *Predications Graph* in which each concept is represented by a node and each relation is a labelled, directed edge from the subject concept to the object concept. Representing the concept-relation space as a graph provides two benefits: As a visual tool, a graph can display the knowledge extracted by the system in a way that is easily understood by the user and can be navigated/explored easily. As a mathematical object, one can employ graph theoretic results when developing algorithms for the reasoning process.

In the work of Wilkowski et al. (2011) an initial graph is constructed by querying a pre-compiled database of predications extracted by SemRep from Medline for all relations containing the a concept. The user then incrementally expands the graph by selecting which terms to query relations for from a list of concepts ranked by their degree centrality (i.e. their degree of connectivity in the graph). After graph construction, potential discovery paths are ranked according to summed degree centrality.

Although some work has been conducted in graph-based LBD, seemingly no research has been conducted on LBD in a global, large-scale predications graph derived from all of Medline, or a sample of it.

2.3.3 Predication-based semantic indexing

Cohen et al. (2012a) propose a hyperdimensional computing technique they call *predication-based semantic indexing* (PSI) for efficient representation and reasoning in the concept-relation space. In PSI, concepts and relations are represented as high-dimensional vectors, where the semantic content of a concept's vector is a combination of all the relations it occurs in and all the concepts it is related to, weighted by the frequency of the relation. The system uses SemRep to extract relations from a sample of 8,182,882 Medline records as input to the training process. Inference in this hyperdimensional space can be performed by ordinary vector operations. The paper shows how PSI enables analogical reasoning along the lines of "x is to what as y is to z?" without explicitly traversing the intermediary relation paths between y and z, leading to efficient inference.

[8] Smalheiser's critique also extends to many of the widely employed co-occurrence based methods. The argument is that research should focus on developing methods that rank interesting relations highly.

The system could originally only infer analogies along a single one of the pathways connecting two concepts x and y. In a later paper Cohen et al. (2012b) expanded the PSI to allow for analogies along multiple pathways, by introducing a vector operation simulating quantum superposition, efficiently reasoning over the entire subgraph connecting x and y. The paper claims that because real world concepts tend to interact through several pathways, literature-based discovery should strive to be able to reason following a similar pattern.

2.4 Approach type hierarchy

From the previous section, it is easy to see the LBD approaches can be divided into a three-level hierarchy according to their dependence on knowledge resources and NLP tools:

Type 1 approaches do not require any knowledge resources: Terms are extracted directly from text, and relations are hypothesized according to co-occurrence patterns. Because all knowledge is extracted directly from text they are completely domain-independent.

Type 2 approaches choose terms from a predefined set of concepts. Co-occurrence patterns are still used to determine relations. The predefined concepts are normally gathered from a domain-specific ontology or vocabulary.

Type 3 approaches use relation extraction tools to extract concepts and relations from text. Because the relations of interest vary widely between domains, domain-specific NLP tools are normally used.

It is evident from the description above that there is a trade-off between reliance on knowledge resources and system performance, as well as a strong correlation between reliance on knowledge resources and domain-dependence. This poses a challenge when adapting LBD approaches to new domains.

3 Domain differences

The current work is a part of a project researching the effects of climate change on the oceanic food web (i.e. who eats who, and how the relative population sizes affect each other) and the biological pump (roughly the ocean's ability to absorb and retain excess atmospheric CO_2). The following section will discuss some of the research issues related to adapting the LBD techniques from the biomedical domain to that of the target domain.

Oceanographic climate science is a cross-disciplinary domain, bringing together researchers from fields such as biology, chemistry, earth science, climate science and oceanography. The cross-disciplinary nature gives rise to an abundance of disjoint literatures, providing strong incentives for LBD. Unfortunately, in a cross-disciplinary domain, scientists from different fields bring their own terminologies and scientific assumptions, creating challenges for LBD work.

While substantial research and engineering effort has gone into the development of NLP tools and computational knowledge sources in the biomedical domain, oceanographic climate science is in this respect under-resourced. To the best of my knowledge, no domain specific NLP tools exist for any sufficiently closely related domain, and although ontologies and controlled vocabularies exist for some of the related disciplines, such as for biology and chemistry, substantial effort is required to identify and combine the desired resources. As a result, it seems unlikely that any of the knowledge intensive (type 2 and 3) LBD methods can be directly applied to oceanographic climate science. Oceanographic climate science also lacks an indexed literature database that covers the entire field, akin to Medline.

Epistemologically there might be a significant difference between the fields: The objects of study (the ocean in oceanographic climate science and the human body in biomedicine) and their processes are quite different, requiring different types of scientific experiments. It therefore seems likely that the structure of the knowledge produced in the different fields might be different. In medicine, experiments can be conducted in a large population of complete systems (human bodies), while in oceanographic experiments must be conducted by sampling subsystems of a single complete system (the ocean). It is therefore not surprising that preliminary observations seem to imply that the results found in oceanographic climate science do not lend themselves to generalization as easily as do those in biomedicine, and that the former have a stronger context dependence (Compare *Eicosapentaenoic acid AFFECTS Vascular constriction* to *Increased labile dissolved organic carbon REDUCES carbon accumulation GIVEN THAT bac-*

teria growth rate is limited). To account for this, text mining tools must be able to extract preconditions as well as relations, or the user must be involved more closely during discovery pattern application to verify that the extracted relations indeed hold true in the same context.

Example discovery patterns for oceanographic climate science have been developed in cooperation with a domain expert, shedding light on some differences between the domains. One research goal of biomedicine is to understand the interactions between domain concepts in order to treat diseases, which is reflected in discovery patterns such as *maybe_treats* (as mentioned in 2.3.1). The discovery patterns developed for oceanographic climate science target the interactions between directional change events (increase or reduce) in quantitative variables, such as *An increase in* CO_2 *causes a decrease in ocean pH*. The types of interactions targeted by these discovery patterns have a more complex structure than the binary relations that define *maybe_treats*. Because most relation extraction tools extract only binary relations, it seems that simply adapting existing relation extraction tools to the domain will not be sufficient.

Ganiz et al. (2006) discusses that LBD lacks a solid theoretic foundation, as most research is applied, rather than theoretical in nature. Although some inquiry has been conducted into the nature of discoveries (Smalheiser, 2012), there is little knowledge about which properties are required to hold in the domain for the LBD methods to be applicable, but the current work assumes that all scientific disciplines are sufficiently similar for LBD methods to be useful.

4 Research directions

The lack of available knowledge resources and NLP tools for the domain makes it hard to directly employ any of the knowledge intensive LBD methods. The development of relation extraction tools for the domain falls outside the scope of the current thesis, and therefore so does the application of type 3 approaches. Instead, the current thesis will focus on bridging the gap between the different terminologies and writing styles caused by different backgrounds in the cross-disciplinary field. To this end, I propose using an unsupervised approach to jointly learn a semantic parser and an ontology from the literature, following the approach of Poon and Domingos (2010).

Poon and Domingos (2009) show that a semantic parser that is able to make non-trivial abstractions from syntactic structure and word usage can be successfully learned in an unsupervised fashion. The system they describe is for instance able to map passive and active form into the same semantic representation and build realistic synonym hierarchies. One challenge that must be addressed is that the current state-of-the-art clusters words based on their argument frames, leading to highly accurate hierarchical clustering of verbs, but lower performance for nouns as these have less diverse argument frames. One research question that will be addressed is how a larger context can be exploited to yield higher performance for nouns.

In an LBD context, the learning process can be seen as bootstrapping a set of concepts for the domain. The resulting system can be considered a hybrid between a type 1 and type 2 approach in terms of the hierarchy defined in 2.4, as it does not use any domain knowledge, but still proposes a set of concepts. A hypothesis that will be evaluated empirically is whether this will provide better results than a pure type 1 system.

The ontology learned by the system can be edited by a domain expert, or combined with ontologies of related fields as they become available, thus providing an elegant interface for integration with domain knowledge in an incremental fashion. The proposed approach will use Markov Logic, a probabilistic extension to first-order logic (FOL), as a knowledge representation language. Background knowledge can therefore easily be incorporated by formulating it as FOL, and the probabilistic aspect enables the system handle contradictions that may occur when combining background knowledge from multiple sources.

The training data set will consist of paper abstracts collected by querying the Mendeley API[9] with a set of keywords that represent the most interesting topics in the domain. The keywords will be developed with the help of a domain expert. As a pre-processing step, the training sentences will be dependency parsed using the Stanford Parser[10]. The proposed LBD system, Houyi[11], will use synonym clusters as concepts, and generate $a \rightarrow b_i$

[9] Mendeley is a web-based reference manager and academic social network that has a large crowd-sourced database of meta-data, such as abstracts, on scientific papers.

[10] `nlp.stanford.edu/software/lex-parser.shtml`

[11] The system is named after a legendary archer in Chinese mythology.

and $b_i \rightarrow c_j$ relation candidates based on td-idf scores. The choice of tf-idf as relation generation/ranking mechanism is motivated by experiments showing that tf-idf gives high recall at the cost of mediocre precision (see section 2.1.5). Because the system is intended to be augmented by relation extraction tools in the future, recall is favoured over precision, as precision is expected to increase in the final version. The discovery candidates are ranked by the number of paths connecting them to a, also motivated by the quantitative experiments described in section 2.1.5.

Houyi will be evaluated quantitatively by comparing performance on a data set divided into training and test data by a cut-off date, following the approach taken by Yetisgen-Yildiz and Pratt (2009). As discussed in section 2.1.5, this is not a perfect evaluation procedure, but it will at least give an indication as to whether unsupervised semantic parsing and ontology building contributes to LBD performance. The baseline system, Sheshou[12], will use the same ranking metric and candidate generation mechanism as Houyi, and uses the NPs extracted by the Stanford Parser as terms.

Development of domain specific ontologies and relation extraction tools is required to apply type 3 LBD methods in the domain. Although outside the scope of the current thesis, it is expected that the resulting semantic parser and ontology can be useful for the development of more sophisticated tools: The semantic parser can function as a preprocessing step for the relation extraction tool by resolving syntactic and synonymic variations. The ontology can be iteratively improved by integrating existing ontologies and human editing, thus providing a point of origin for domain knowledge engineering.

Acknowledgements

This paper is a part of an ongoing master's thesis under the supervision of Pinar Öztürk, with Erwin Marsi as co-supervisor. I am extremely grateful for their feedback and support.

References

Alan R. Aronson and François-Michel M. Lang. 2010. An overview of MetaMap: historical perspective and recent advances. *Journal of the American Medical Informatics Association : JAMIA*, 17(3):229–236, May.

Delroy Cameron, Olivier Bodenreider, Hima Yalamanchili, Tu Danh, Sreeram Vallabhaneni, Krishnaprasad Thirunarayan, Amit P. Sheth, and Thomas C. Rindflesch. 2013. A graph-based recovery and decomposition of swanson's hypothesis using semantic predications. *J. of Biomedical Informatics*, 46(2):238–251, April.

Trevor Cohen, Dominic Widdows, Roger W. Schvaneveldt, Peter Davies, and Thomas C. Rindflesch. 2012a. Discovering discovery patterns with predication-based Semantic Indexing. *Journal of Biomedical Informatics*, 45(6):1049–1065, December.

Trevor Cohen, Dominic Widdows, Lance Vine, Roger Schvaneveldt, and Thomas C. Rindflesch. 2012b. Many Paths Lead to Discovery: Analogical Retrieval of Cancer Therapies. In *Quantum Interaction*, volume 7620 of *Lecture Notes in Computer Science*, pages 90–101. Springer Berlin Heidelberg.

Ralph A. Digiacomo, Joel M. Kremer, and Dhiraj M. Shah. 1989. Fish-oil dietary supplementation in patients with Raynaud's phenomenon: A double-blind, controlled, prospective study. *The American Journal of Medicine*, 86(2):158–164, January.

Murat C. Ganiz, William M. Pottenger, and Christopher D. Janneck. 2006. Recent Advances in Literature Based Discovery. In *Journal of the American Society for Information Science and Technology*.

Michael D. Gordon and Susan Dumais. 1998. Using latent semantic indexing for literature based discovery. *Journal of the American Society for Information Science and Technology*, 49(8):674–685, June.

Michael D. Gordon and Robert K. Lindsay. 1996. Toward discovery support systems: a replication, re-examination, and extension of Swanson's work on literature-based discovery of a connection between Raynaud's and fish oil. *Journal of the American Society for Information Science and Technology*, 47(2):116–128, February.

Michael Gordon, Robert K. Lindsay, and Weiguo Fan. 2001. Literature Based Discovery on the World Wide Web. In *ACM Transactions on Internet Technology*, pages 261–275, New York, USA. ACM Press.

Dimitar Hristovski, J. Stare, B. Peterlin, and S. Dzeroski. 2001. Supporting discovery in medicine by association rule mining in Medline and UMLS. *Studies in health technology and informatics*, 84(Pt 2):1344–1348.

Dimitar Hristovski, Carol Friedman, Thomas C. Rindflesch, and Borut Peterlin. 2006. Exploiting semantic relations for literature-based discovery. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, pages 349–353.

[12] Mandarin Chinese for "archer", a reference to Arrowsmith.

Dimitar Hristovski, C. Friedman, T. C. Rindflesch, and B. Peterlin. 2008. Literature-Based Knowledge Discovery using Natural Language Processing. In Peter Bruza and Marc Weeber, editors, *Literature-based Discovery*, volume 15 of *Information Science and Knowledge Management*, chapter 9, pages 133–152. Springer, Heidelberg, Germany.

Ronald N. Kostoff. 2007. Validating discovery in literature-based discovery. *Journal of Biomedical Informatics*, 40(4):448–450, August.

Robert K. Lindsay and Michael D. Gordon. 1999. Literature-based discovery by lexical statistics. *Journal of the American Society for Information Science*, pages 574–587.

Hoifung Poon and Pedro Domingos. 2009. Unsupervised semantic parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 1–10, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hoifung Poon and Pedro Domingos. 2010. Unsupervised ontology induction from text. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 296–305, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wanda Pratt and Meliha Yetisgen-Yildiz. 2003. LitLinker: capturing connections across the biomedical literature. In *Proceedings of the 2nd international conference on Knowledge capture*, K-CAP '03, pages 105–112, New York, NY, USA. ACM.

Thomas C. Rindflesch and Marcelo Fiszman. 2003. The interaction of domain knowledge and linguistic structure in natural language processing: interpreting hypernymic propositions in biomedical text. *Journal of Biomedical Informatics*, 36(6):462–477, December.

M. J. Schuemie, M. Weeber, B. J. Schijvenaars, E. M. van Mulligen, C. C. van der Eijk, R. Jelier, B. Mons, and J. A. Kors. 2004. Distribution of information in biomedical abstracts and full-text publications. *Bioinformatics*, 20(16):2597–2604, November.

Neil R. Smalheiser. 2012. Literature-based discovery: Beyond the ABCs. *Journal of the American Society for Information Science and Technology*, 63(2):218–224, February.

Don R. Swanson and Neil R. Smalheiser. 1997. An interactive system for finding complementary literatures: a stimulus to scientific discovery. *Artificial Intelligence*, 91(2):183–203, April.

Don R. Swanson. 1986. Undiscovered public knowledge. *The Library Quarterly*, 56(2):pp. 103–118.

Don R. Swanson. 1988. Migraine and magnesium: eleven neglected connections. *Perspectives in Biology and Medicine*, 31(4):526–557.

Don R. Swanson. 1991. Complementary structures in disjoint science literatures. In *SIGIR '91: Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 280–289, New York, NY, USA. ACM.

Marc Weeber, Henny Klein, Lolkje T. de Jong van den Berg, and Rein Vos. 2001. Using concepts in literature-based discovery: Simulating Swanson's Raynaud-fish oil and migraine-magnesium discoveries. *Journal of the American Society for Information Science and Technology*, 52(7):548–557.

Bartłomiej Wilkowski, Marcelo Fiszman, Christopher M. Miller, Dimitar Hristovski, Sivaram Arabandi, Graciela Rosemblat, and Thomas C. Rindflesch. 2011. Graph-based methods for discovery browsing with semantic predications. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, 2011:1514–1523.

Jonathan D. Wren, Raffi Bekeredjian, Jelena A. Stewart, Ralph V. Shohet, and Harold R. Garner. 2004. Knowledge discovery by automated identification and ranking of implicit relationships. *Bioinformatics (Oxford, England)*, 20(3):389–398, February.

Jonathan D. Wren. 2004. Extending the mutual information measure to rank inferred literature relationships. *BMC bioinformatics*, 5, October.

Meliha Yetisgen-Yildiz and Wanda Pratt. 2006. Using statistical and knowledge-based approaches for literature-based discovery. *Journal of Biomedical Informatics*, 39(6):600–611, December.

Meliha Yetisgen-Yildiz and Wanda Pratt. 2009. A new evaluation methodology for literature-based discovery systems. *Journal of biomedical informatics*, 42(4):633–643, August.

Unsupervised Relation Extraction of In-Domain Data from Focused Crawls

Steffen Remus

FG Language Technology
Computer Science Department, Technische Universität Darmstadt

Information Center for Education
German Institute for Educational Research (DIPF)

remus@cs.tu-darmstadt.de

Abstract

This thesis proposal approaches unsupervised relation extraction from web data, which is collected by crawling only those parts of the web that are from the same domain as a relatively small reference corpus. The first part of this proposal is concerned with the efficient discovery of web documents for a particular domain and in a particular language. We create a combined, focused web crawling system that automatically collects relevant documents and minimizes the amount of irrelevant web content. The collected web data is semantically processed in order to acquire rich in-domain knowledge. Here, we focus on fully unsupervised relation extraction by employing the extended distributional hypothesis. We use distributional similarities between two pairs of nominals based on dependency paths as context and vice versa for identifying relational structure. We apply our system for the domain of educational sciences by focusing primarily on crawling scientific educational publications in the web. We are able to produce promising initial results on relation identification and we will discuss future directions.

1 Introduction

Knowledge acquisition from written or spoken text is a field of interest not only for theoretical reasons but also for practical applications, such as semantic search, question answering and knowledge management, just to name a few.

In this work, we propose an approach for *unsupervised relation extraction* (URE) where we make use of the *Distributional Hypothesis* by Harris (1954). The underlying data set is collected from the world wide web by focusing on web documents that are from the same domain as a small initialization data set that is provided beforehand. We hereby enrich this existing, domain-defining, corpus with more data of the same kind. This is needed for practical reasons when working with the Distributional Hypothesis (Harris, 1954): A lot of data is required for plausible outcomes and an appropriate coverage. However, we want as little irrelevant data as possible. The proposal's contribution is thus twofold: *a*) focused crawling, and *b*) unsupervised relation extraction. As a particular use case, we are especially interested in scientific publications from the German educational domain. However, we would like to point out that the methodology itself is independent of language and domain and is generally applicable to any domain.

This work is structured as follows: First we will motivate our combined approach and introduce each part individually. We then present related work in Section 2. Section 3 explains the methodology of both parts, and in Section 4 we outline the evaluation procedure of each of the components individually. This is followed by some preliminary results in Section 5, and Section 6 concludes this proposal with some prospects for future work.

1.1 Motivation

The identification of relations between entities solely from text is one of many challenges in the development of language understanding system (Carlson et al., 2010; Etzioni et al., 2008); and yet it is the one step with the highest information gain. It is used e.g. for taxonomy induction (Hearst, 1992) or ontology accumulation (Mintz et al., 2009) or even for identifying facts that express general knowledge and that often recur (Chambers and Jurafsky, 2011). Davidov et al. (2007) performed unsupervised relation extraction by actively mining the web and showed major improve-

Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 11–20,
Gothenburg, Sweden, April 26-30 2014. ©2014 Association for Computational Linguistics

ments in the detection of new facts from only little initial seed. They used a major web search engine as a vital component of their system. According to Kilgarriff (2007), however, this strategy is unreliable and should be avoided. Nevertheless, the web is undeniably the largest source for any kind of data, and we feel the need for developing easy-to-use components that make it possible to create corpora from the web with only little effort (cf. e.g. Biemann et al. (2013)). When it comes to specific in-domain information, the complete world wide web is first of all too vast to be processed conveniently, and second the gain is little because of too much irrelevant information. Thus we need methods for reducing the size of data to process without losing the focus on the important information and without using web search engines. The combination of a focused crawling system with a subsequent unsupervised relation extraction system enables the acquisition of richer in-domain knowledge than just relying on little local data, but without having to process petabytes of data and still not relying of web search. And yet, by using the web as a resource, our system is generally applicable and independent of language and target domain.

1.2 Focused Crawling

The first part of this proposal is concerned with the efficient discovery of publications in the web for a particular domain. The domain definition is given as a limited number of reference documents. An extra challenge is, that non-negligible amounts of scientific publications are only available as pdf documents, which makes the necessity of new focused crawling techniques even more important. This holds especially for our target use case, the German educational domain. In Section 2.1 we will discuss this issue in more detail. We develop a focused web crawling system which collects primarily relevant documents and ignores irrelevant documents and which is particularly suited for harvesting documents from a predefined specific domain.

1.3 Unsupervised Relation Extraction

The second part of this proposal is the semantic structuring of texts — in our particular use case scientific publications from the educational domain — by using data-driven techniques of computational semantics. The resulting structure enables forms of post-processing like inference or reasoning. In the semantic structuring part, the overall goal is to discover knowledge which can then be used in further steps. Specifically, we will focus on unsupervised relation extraction.

2 Related Work

2.1 Focused Crawling

The development of high-quality data-driven semantic models relies on corpora of large sizes (Banko and Brill, 2001; Halevy et al., 2009), and the world wide web is by far the biggest available source of textual data. Nowadays, a large number of research projects rely on corpora that comes from data in the world wide web. The Web-as-Corpus Kool Yinitiative[1] (WaCKy) (Baroni et al., 2009) for example produced one of the largest corpora used in linguistic research which comes from web documents. Another research initiative which produces a variety of corpora by crawling the web is the COW[2] (corpora from the web) project (Schäfer and Bildhauer, 2012). Currently one of the largest N-gram corpora coming from web data is the Google V1 and Google V2 (Lin et al., 2010), which are used e.g. for improving noun phrase parsing (Pitler et al., 2010). Also the predecessor Google Web1T (Brants and Franz, 2006), which is computed from 1 Trillion words from the web, is heavily used in the community.

All these corpora are generated from general texts which either come from crawling specific *top-level-domains* (tlds) or preprocessing and filtering very large amounts of texts for a specified language. Additionally, we are not aware of any corpus that is created by collecting pdf documents. This is especially an issue when aiming at a corpus of scientific publications, such as e.g. the ACL anthology[3] (Bird et al., 2008). As of today, electronic publications are primarily distributed as pdf documents. Usually these are omitted by the particular crawler because of a number of practical issues, e.g. difficulties in extracting clean plaintext.

Further, we are not interested in sheer collection size, but also in domain specificity. Crawling is a time-consuming process and it comes with logistic challenges for processing the resulting data. While standard breadth-first or depth-first crawling strategies can be adjusted to include pdf files, we want to avoid to harvest the huge bulk of data

[1] http://wacky.sslmit.unibo.it/

[2] http://hpsg.fu-berlin.de/cow/

[3] http://acl-arc.comp.nus.edu.sg/

that we are not interested in, namely those documents that are of a different topical domain as our initial domain definition.

In focused crawling, which is sometimes also referred to as topical crawling, web crawlers are designed to harvest those parts of the web first that are more interesting for a particular topic (Chakrabarti et al., 1999). By doing so, task-specific corpora can be generated fast and efficient. Typical focused crawlers use machine learning techniques or heuristics to prioritize newly discovered URIs (unified resource identifier) for further crawling (Blum and Mitchell, 1998; Chakrabarti et al., 1999; Menczer et al., 2004). In our scenario however, we do not rely on positively and negatively labeled data. The source documents that serve as the domain definition are assumed to be given in plain text. The development of tools that are able to generate in-domain web-corpora from focused crawls is the premise for further generating rich semantic models tailored to a target domain.

2.2 Unsupervised Relation Extraction

The unsupervised relation extraction (URE) part of this proposal is specifically focused on extracting relations between *nominals*. Typically the choice of the entity type depends merely on the final task at hand. Kinds of entities which are usually considered in relation extraction are named entities like persons or organizations. However, we will focus on nominals which are much more general and also include named entities since they are basically nouns or noun phrases (Nastase et al., 2013). Nominals are discussed in more detail in Section 3.2. Unsupervised methods for relation extraction is a particularly interesting area of research because of its applicability across languages without relying on labeled data. In contrast to *open information extraction*, in unsupervised relation extraction the collected relations are aggregated in order to identify the most promising relations for expressing interesting facts. Here, the grouping is made explicit for further processing.

One possible application of relation extraction is the establishment of so-called *knowledge graphs* (Sowa, 2000), which encode facts that manifest solely from text. The knowledge graph can then be used e.g. for reasoning, that is finding new facts from existing facts.

Many approaches exist for acquiring knowledge from text. Hearst (1992) first discovered that relations between entities occur in a handful of well developed text patterns. For example '*X is a Y*' or '*X and other Ys*' manifest themselves as hyponymic relations. However, not every kind of relation is as easy to identify as those '*is-a*' relations. Often semantic relations cannot be expressed by any pattern. A variety of methods were developed that automatically find new patterns and entities with or without supervision. These methods reach from *bootstrapping methods* (Hearst, 1992) over *distant supervision* (Mintz et al., 2009) and *latent relational analysis* (LRA) (Turney, 2005) to *extreme unsupervised relation extraction* (Davidov and Rappoport, 2008a), just to name a few. The importance of unsupervised methods for relation extraction is obvious: The manual creation of knowledge resources is time consuming and expensive in terms of manpower. Though manual resources are typically very precise they are almost always lacking of lexical and relational coverage.

The extraction of relations between entities is a crucial process which is performed by every modern language understanding system like NELL[4] (Carlson et al., 2010) or machine reading[5], which evolved among others from TextRunner[6] (Etzioni et al., 2008). The identification of relations in natural language texts is at the heart of such systems.

3 Methodology

3.1 Focused Crawling

Language models (LMs) are a rather old but well understood and generally accepted concept in Computational Linguistics and Information Retrieval. Our focused crawling strategy builds upon the idea of utilizing a language model to discriminate between relevant and irrelevant web documents. The key idea of this methodology is that web pages which come from a certain domain — which implies the use of a particular vocabulary (Biber, 1995) — link to other documents of the same domain. The assumption is that the crawler will most likely stay in the same topical domain as the initial language model was generated from. Thus the crawling process can be terminated when enough data has been collected.

[4] Never Ending Language Learner: `http://rtw.ml.cmu.edu/`

[5] `http://ai.cs.washington.edu/projects/open-information-extraction`

[6] `http://openie.cs.washington.edu/`

A language model is a statistical model over short sequences of consecutive tokens called N-grams. The order of a language model is defined by the length of such sequences, i.e. the 'N' in N-gram. The probability of a sequence of m words, that could be for example a sentence, is computed as:

$$p(w_1, ..., w_m) \approx \prod_{i=1}^{m} p(w_i | w_{i-N+1:i-1}) \ , \qquad (1)$$

where N is the order of the language model and $p(w_i | w_{i-n+1:i-1})$ is the probability of the particular N-gram. In the simplest case the probability of an N-gram is computed as:

$$p(w_i | w_{i-n+1:i-1}) = \frac{count(w_{i-N+1:i})}{count(w_{i-N+1:i-1})} \ , \qquad (2)$$

where $count$(N-gram) is a function that takes as argument an N-gram of length N or an N-gram of length $N - 1$ and returns the frequency of observations in the source corpus. This model has some obvious limitations when it comes to *out-of-vocabulary* (OOV) terms because of probabilities being zero. Due to this limitation, a number of LMs were proposed which handle OOV terms well.

One of the most advanced language models is the Kneser-Ney language model (Kneser and Ney, 1995), which applies an advanced interpolation technique for OOV issues. According to Halevy et al. (2009), simpler models that are trained on large amounts of data often outperform complex models with training procedures that are feasible only for small data. Anyway, we have only little data in the initial phase, thus we use Kneser and Ney's model.

Perplexity is used to measure the amount of compatibility with another model X:

$$Perplexity(X) = 2^{H(X)} \ , \qquad (3)$$

where $H(X) = -\frac{1}{|X|} \sum_{x \in X} \log_2 p(x)$ is the cross entropy of a model X. Using perplexity we are able to tell how well the language model fits the data and vice versa.

The key idea is that documents which come from a certain register or domain — which implies the use of a particular vocabulary (Biber, 1995) — link to other documents of the same register. Using perplexity, we are able to rank outgoing links by their deviation from our initial language model. Hence weblinks that are extracted from a highly deviating webpage are less prioritized for harvesting. The open source crawler software Heritrix[7] (Mohr et al., 2004) forms the basis of our focused crawling strategy, since it provides a well-established framework which is easily extensible through its modularity.

3.2 Identification of Nominals

Nominals are defined to be expressions which syntactically act like nouns or noun phrases (Quirk et al., 1985, p.335). Another definition according to Nastase et al. (2013) is that nominals are defined to be in one of the following classes: *a*) common nouns, *b*) proper nouns, *c*) multi-word proper nouns, *d*) deverbal nouns, *e*) deadjectival nouns, or *f*) non-compositional (adjective) noun phrases. In this work we will follow the definition given by Nastase et al. (2013). We will further address only relations that are at least realized by verbal or prepositional phrases and ignore relations that are implicitly present in compounds, which is a task of its own, cf. (Holz and Biemann, 2008). Note however we do not ignore relations between compounds, but within compounds.

The identification of nominals can be seen as the task of identifying reliable *multi-word-expressions* (MWEs), which is a research question of its own right. As a first simplified approach we only consider nouns and heads of noun compounds to be representatives for nominals. E.g. a compound is used as an entity, but only the head is taken into further consideration as a representative since it encapsulates the main meaning for that phrase.

3.3 Unsupervised Relation Extraction

Our system is founded in the idea of distributional semantics on the level of dependency parses. The *Distributional Hypothesis* by Harris (1954) (cf. also (Miller and Charles, 1991)) states that words which tend to occur in similar contexts tend to have similar meanings. This implies that one can estimate the meaning of an unknown word by considering the context in that it occurs. Lin and Pantel (2001) extended this hypothesis to cover shortest paths in the dependency graph — so-called dependency paths — and introduced the *Extended Distributional Hypothesis*. This extended hypothesis states that dependency paths which tend to occur in similar contexts, i.e. they connect the simi-

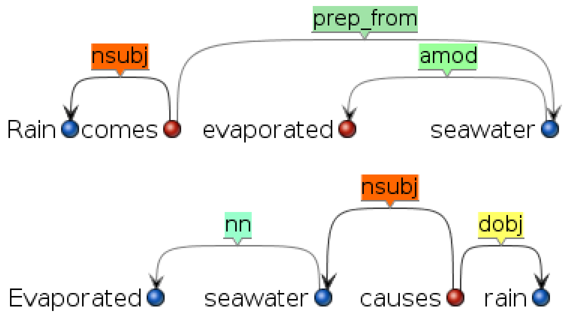
[7] `http://crawler.archive.org`

lar sets of words, also tend to have similar meanings.

Sun and Grishman (2010) used an agglomerative hierarchical clustering based approach in order to group the patterns found by Lin and Pantel's method. The clusters are used in a semi-supervised way to extract relation instances that are used in a bootstrapping fashion to find new relations. While Sun and Grishman (2010) performed a hard clustering, meaning every relation is assigned exactly to one cluster, we argue that relations are accompanied by a certain degree of ambiguity. Think for example about the expression '*X comes from Y*' which could be both, a causal relation or a locational relation depending on the meaning of *X* and *Y*.

That being said, we use the Extended Distributional Hypothesis in order to extract meaningful relations from text. We follow Lin and Pantel (2001) and use the dependency path between two entities to identify both, similar entity pairs and similar dependency paths. Specifically we use the Stanford Parser[8] (Klein and Manning, 2003) to get a collapsed dependency graph representation of a sentence, and apply the JoBimText[9] (Biemann and Riedl, 2013) software for computing the distributional similarities.

By using the JoBimText framework, we accept their theory, which states that dimensionality-reduced vector space models are not expressive enough to capture the full semantics of words, phrases, sentences, documents or relations. Turney and Pantel (2010) surveyed that vector space models are commonly used in computational semantics and that they are able to capture the meaning of words. However, by doing various kinds of vector space transformations, e.g. dimensionality reduction with SVD[10] important information from the long tail, i.e. items that do not occur often, is lost. Instead, Biemann and Riedl (2013) introduced the scalable JoBimText framework, which makes use of the Distributional Hypothesis. We take this as a starting point to steer away from the use of vector space models.



$$\begin{array}{ll}
\mathit{rain} :: \mathit{seawater} & @1 \xleftarrow{nsubj} \mathit{comes} \xrightarrow{prep\_from} @2 \\
\mathit{rain} :: \mathit{seawater} & @1 \xleftarrow{dobj} \mathit{causes} \xrightarrow{nsubj} @2 \\
\mathit{seawater} :: \mathit{rain} & @1 \xleftarrow{prep\_from} \mathit{comes} \xrightarrow{nsubj} @2 \\
\mathit{seawater} :: \mathit{rain} & @1 \xleftarrow{nsubj} \mathit{causes} \xrightarrow{dobj} @2
\end{array}$$

Figure 1: Upper[12]: collapsed dependency parses of the example sentences '*Rain comes from evaporated seawater.*' and *'Evaporated seawater causes rain'*. Lower: extracted entity pairs plus shortest dependency paths per entity pair from both sentences.

For each entity pair 'X::Y', where 'X' and 'Y' are nominals, we collect all dependency paths that co-occur with it in the complete dataset. A particular path for a particular relation instance has form '@1-PATH-@2', where '-PATH-' is the instantiation of the directed shortest path in the collapsed dependency path starting from a particular 'X' and ending in a particular 'Y'. The @1, resp. @2, symbolizes the place where 'X' and 'Y' were found in the path. Here we restrict the path to be shorter than five edges and additionally we ignore paths that have only *nn* relations, i.e. compound dependency relations. See Figure 1 for an illustration of this strategy on two small example sentences. Note that this procedure strongly coheres with the methodologies proposed by Lewis and Steedman (2013) or Akbik et al. (2013).

We then compute the distributional similarities for both directions: *a*) similarities of entity pairs by paths, and *b*) similarities of paths by entity pairs. This gives us two different views on the data.

4 Evaluation

The two major directions of this paper, i.e. the focused crawling part and the unsupervised relation extraction part are evaluated individually and independent of each other. First we will present an

[8] `http://nlp.stanford.edu/downloads/lex-parser.shtml`

[9] `http://sf.net/p/jobimtext`

[10] Singular Value Decomposition, used for example in latent semantic analysis, latent relational analysis, principal component analysis and many more.

[12] Images generated with GrammarScope: `http://grammarscope.sf.net` .

evaluation methodology to assess the quality of the crawler and second we will outline the evaluation of relations. While we can only show anecdotical evidence of the viability of this approach, since the work is in progress, we are able to present encouraging preliminary results in Section 5.

4.1 Focused Crawling

The quality of a focused crawl is measured in terms of perplexity (cf. Section 3.1) by creating a language model from the harvested data during a particular crawl. Perplexity is then calculated with respect to a held out test set. The following three phases describe the evaluation procedure more precisely:

1. The source corpus is split i.i.d.[13] into a training and test set.

2. We create a language model U of the training data, which is applied according to Section 3.1 for automatically focusing the crawl. In order to compare the data of different crawls, the repeated crawls are initialized with the same global parameter settings, e.g. politeness settings, seed, etc. are the same, and are terminated after reaching a certain number of documents.

3. From the harvested data, another language model V is produced which is used for the evaluation of the test data. Here we argue that a crawl which collects data that is used for evaluating V and V results in a lower perplexity score, is preferred as it better models the target domain.

Figure 2 shows a schematic overview of the three phases of evaluation.

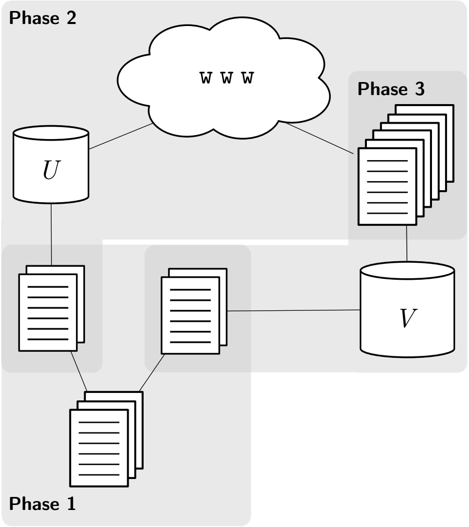


Figure 2: Schematic overview of the evaluation procedure for a particular crawl.

4.2 Unsupervised Relation Extraction

The evaluation of relation extraction is a non-trivial task, as unsupervised categories do usually not exactly match the distinctions taken in annotation studies. For the evaluation of our method we consider the following three approaches:

1. We test our relations directly on datasets that were provided as relation classification challenge datasets (Girju et al., 2007; Hendrickx et al., 2010). Whereas the first dataset is provided as a binary classification task, the second is a multi-way classification task. However, both datasets can be transformed to address the one or the other task. This is possible because the challenge is already finished.

2. We apply our extracted relations for assisting classification algorithms for the task of *textual entailment* (Dagan et al., 2006).

3. Following Davidov and Rappoport (2008b) we would further like to apply our system to the task of question answering.

While the first approach is an intrinsic evaluation, the other three approaches are extrinsic, i.e. the extracted relations are used in a particular task which is then evaluated against some gold standard.

5 Preliminary Results

5.1 Focused crawling

Table 1 shows some quantitative characteristics of a non-focused crawl. Here the crawl was performed as a *scoped crawl*, which means that it was bounded to the German top-level-domain *'.de'* and additionally by a maximum number of 20 hops from the start seed[14]. The crawl was terminated after about two weeks. Although these numbers

[13] independent and identically distributed

[14] The start seed for the first crawl consists of five web page urls which are strongly connected to German educational research.

	pdf	html
size in GBytes	17	400
number of documents	43K	9M
runtime	≈ 2 weeks	

Table 1: Numbers are given as approximate numbers.

do not seem surprising, they do support the main argument of this proposal. Focused crawling is necessary in order to reduce the massive load of irrelevant data.

Initial encouraging results on the comparison of a focused vs. a non-focused crawl are shown in Figure 3. The crawls were performed under the same conditions and we recorded the perplexity value during the process. We plot the history for the first 300,000 documents. Although these results are preliminary, a trend is clearly observable. The focused crawl harvests more relevant documents as it proceeds, whereas the non-focused crawl deviates more as longer the crawl proceeds, as indicated by higher perplexity values for later documents — an effect that is likely to increase as the crawl proceeds. The focused crawl, on the other hand, stays within low perplexity limits. We plan to evaluate settings and the interplay between crawling parameters and language modeling more thoroughly in future evaluations.

5.2 Unsupervised Relation Extraction

The unsupervised extraction of relations was performed on a small subset of one Million sentences of the news corpus from the Leipzig Corpora Collection (Richter et al., 2006).

Preliminary example results are shown in Table 2 and in Table 3. Table 2 shows selected results for similar entity pairs, and Table 3 shows selected results for similar dependency paths.

In Table 2, three example entity pairs are shown together with their most similar counterparts. It is interesting to see that the relation of *gold* to *ounce* is the same as *stock* to *share* or *oil* to *barrel* and we can easily agree here, since the one is the measuring unit for the other.

Table 3 shows for three example prepositional paths the similar paths. We have chosen prepositional phrases here because of their intuitive interpretability. The example output shows that the similar phrases which were identified by the system are also interpretable for humans.

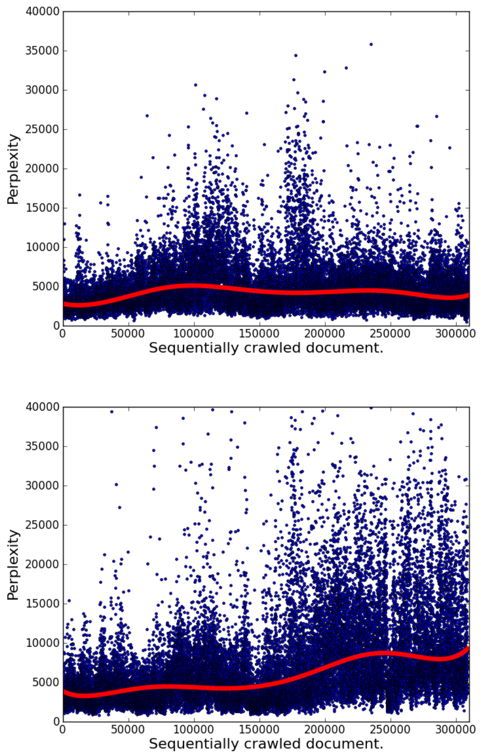


Figure 3: Two crawl runs under same conditions and with same settings. Upper: a focused crawl run. Lower: a non-focused crawl run.

6 Conclusion and Future Work

This research thesis proposal addressed the two major objectives:

1. crawling with a focus on in-domain data by using a language model of an initial corpus, which is small compared to the expected result of the crawls, in order to discriminate relevant web documents from irrelevant web documents, and

2. unsupervised relation extraction by following the principles of the Distributional Hypothesis by Harris (1954) resp. the Extended Distributional Hypothesis by Lin and Pantel (2001).

The promising preliminary results encourage us to examine this approach for further directions. Specifically the yet unaddressed parts of the evaluation will be investigated. Further, the unsupervised relation extraction techniques will be applied on the complete set of in-domain data, thus finalizing the workflow of enriching a small amount of domain defining data with web data

gold/NN :: ounce/NN
crude/NN :: barrel/NN
oil/NN :: barrel/NN
futures/NNS :: barrel/NN
stock/NN :: share/NN
graduate/NN :: University/NNP
graduate/NN :: School/NNP
graduate/NN :: College/NNP
goals/NNS :: season/NN
points/NNS :: season/NN
points/NNS :: game/NN
touchdowns/NNS :: season/NN

Table 2: Example results for selected entity pairs. Similar entity pairs with respect to the boldface pair are shown.

@1 <= prep_above = @2
@1 <= *prep_below* = @2
@1 <= *nsubj* = *rose/VBD* = *dobj* => @2
@1 <= *nsubj* = *dropped/VBD* = *dobj* => @2
@1 <= *nsubj* = *fell/VBD* = *dobj* => @2
@1 <= prep_regarding = @2
@1 <= *prep_about* = @2
@1 <= *prep_on* = @2
@1 <= prep_like = @2
@1 <= *prep_such_as* = @2
@1 <= *prep_including* = @2
@1 <= *nsubj* = *are/VBP* = *prep_among* => @2

Table 3: Example results for selected dependency paths. Similar paths with respect to the boldface path are shown.

from focused crawls in order to extract rich in-domain knowledge, particularly from the german educational domain as our application domain. While we made clear that crawling the web is a crucial process in order to get the amounts of in-domain data needed by the unsupervised relation extraction methods, we did not yet point out that we will also examine the reverse direction, i.e. the possibility to use the extracted relations for further improving the focused crawler. A focused crawler that is powered by semantic relations between entities would raise a new level of semantically focused crawls. Additionally, we will investigate possibilities for further narrowing the relations found by our system. Here it is possible to further categorize or cluster the relations by using either the similarity graph or the features itself, as done by Pantel and Lin (2002).

Acknowledgments

This work has been supported by the German Institute for Educational Research (DIPF) under the KDSL program.

References

Alan Akbik, Larysa Visengeriyeva, Johannes Kirschnick, and Alexander Löser. 2013. Effective selectional restrictions for unsupervised relation extraction. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing (IJCNLP)*, pages 1312–1320, Nagoya, Japan.

Michele Banko and Eric Brill. 2001. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics (ACL)*, pages 26–33, Toulouse, France.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

Douglas Biber. 1995. *Dimensions of Register Variation: A Cross-Linguistic Comparison.* Cambridge University Press, Cambridge.

Chris Biemann and Martin Riedl. 2013. Text: Now in 2d! a framework for lexical expansion with contextual similarity. *Journal of Language Modelling (JLM)*, 1(1):55–95.

Chris Biemann, Felix Bildhauer, Stefan Evert, Dirk Goldhahn, Uwe Quasthoff, Roland Schäfer, Johannes Simon, Swiezinski Swiezinski, and Torsten Zesch. 2013. Scalable construction of high-quality web corpora. *Journal for Language Technology and Computational Linguistics (JLCL)*, 27(2).

Steven Bird, Robert Dale, Bonnie Dorr, Bryan Gibson, Mark Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir Radev, and Yee Fan Tan. 2008. The acl anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*, Marrakech, Morocco.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory (COLT)*, pages 92–100, Madison, Wisconsin, USA.

Thorsten Brants and Alex Franz. 2006. *Web 1T 5-gram Version 1*. Linguistic Data Consortium, Philadelphia.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the 24th Conference on Artificial Intelligence (AAAI)*, Atlanta, GA, USA.

Soumen Chakrabarti, Martin van den Berg, and Byron Dom. 1999. Focused crawling: a new approach to topic-specific web resource discovery. *Computer Networks*, 31(11–16):1623–1640.

Nathanael Chambers and Dan Jurafsky. 2011. Template-based information extraction without the templates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 976–986, Portland, Oregon, USA.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, volume 3944 of *Lecture Notes in Computer Science*, pages 177–190. Springer Berlin Heidelberg.

Dmitry Davidov and Ari Rappoport. 2008a. Classification of semantic relationships between nominals using pattern clusters. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 227–235, Columbus, Ohio.

Dmitry Davidov and Ari Rappoport. 2008b. Unsupervised discovery of generic relationships using pattern clusters and its evaluation by automatically generated SAT analogy questions. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 692–700, Columbus, Ohio.

Dmitry Davidov, Ari Rappoport, and Moshe Koppel. 2007. Fully unsupervised discovery of concept-specific relationships by web mining. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 232–239, Prague, Czech Republic.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. 2008. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.

Roxana Girju, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2007. Semeval-2007 task 04: Classification of semantic relations between nominals. In *Proceedings of the Fourth International Workshop on Semantic Evaluation (SemEval)*, pages 13–18, Prague, Czech Republic.

Alon Halevy, Peter Norvig, and Fernando Pereira. 2009. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12.

Zellig S. Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 15th Conference on Computational Linguistics (Coling)*, pages 539–545, Nantes, France.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Fifth International Workshop on Semantic Evaluation (SemEval)*, pages 33–38, Los Angeles, California.

Florian Holz and Chris Biemann. 2008. Unsupervised and knowledge-free learning of compound splits and periphrases. In *CICLing 2008: Proceedings of the Conference on Intelligent Text Processing and Computational Linguistics*, pages 117–127, Haifa, Israel.

Adam Kilgarriff. 2007. Googleology is bad science. *Computational Linguististics (CL)*, 33(1):147–151.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL)*, pages 423–430, Sapporo, Japan.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 181–184, Detroit, Michigan.

Mike Lewis and Mark Steedman. 2013. Unsupervised induction of cross-lingual semantic relations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 681–692, Seattle, WA, USA.

Dekang Lin and Patrick Pantel. 2001. Dirt - discovery of inference rules from text. In *Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 323–328, San Francisco, California.

Dekang Lin, Kenneth Church, Heng Ji, Satoshi Sekine, David Yarowsky, Shane Bergsma, Kailash Patil, Emily Pitler, Rachel Lathbury, Vikram Rao, Kapil Dalwani, and Sushant Narsale. 2010. New tools for web-scale n-grams. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC)*, pages 2221–2227, Valletta, Malta.

Filippo Menczer, Gautam Pant, and Padmini Srinivasan. 2004. Topical web crawlers: Evaluating adaptive algorithms. *ACM Transactions Internet Technology (TOIT)*, 4(4):378–419.

George A Miller and Walter G Charles. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes (LCP)*, 6(1):1–28.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore.

Gordon Mohr, Michele Kimpton, Micheal Stack, and Igor Ranitovic. 2004. Introduction to heritrix, an archival quality web crawler. In *Proceedings of the 4th International Web Archiving Workshop IWAW'04*, Bath, UK.

Vivi Nastase, Preslav Nakov, Diarmuid Ó Séaghdha, and Stan Szpakowicz. 2013. Semantic relations between nominals. In *Synthesis Lectures on Human Language Technologies*, volume 6. Morgan & Caypool Publishers.

Patrick Pantel and Dekang Lin. 2002. Document clustering with committees. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 199–206.

Emily Pitler, Shane Bergsma, Dekang Lin, and Kenneth Church. 2010. Using web-scale n-grams to improve base np parsing performance. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling)*, pages 886–894, Beijing, China.

Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A Comprehensive Grammar of the English Language*. Longman, London.

Matthias Richter, Uwe Quasthoff, Erla Hallsteinsdóttir, and Chris Biemann. 2006. Exploiting the leipzig corpora collection. In *Proceesings of the IS-LTC*, Ljubljana, Slovenia.

Roland Schäfer and Felix Bildhauer. 2012. Building large corpora from the web using a new efficient tool chain. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*, pages 486–493, Istanbul, Turkey.

John Sowa. 2000. *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Brooks Cole Publishing Co., Pacific Grove, CA.

Ang Sun and Ralph Grishman. 2010. Semi-supervised semantic pattern discovery with guidance from unsupervised pattern clusters. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling)*, pages 1194–1202, Beijing, China.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal for Artificial Intelligence Research (JAIR)*, 37:141–188.

Peter D. Turney. 2005. Measuring semantic similarity by latent relational analysis. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1136–1141, Edinburgh, Scotland, UK.

Enhancing Medical Named Entity Recognition with Features Derived from Unsupervised Methods

Maria Skeppstedt
Dept. of Computer and Systems Sciences (DSV)
Stockholm University, Forum 100, 164 40 Kista, Sweden
mariask@dsv.su.se

Abstract

A study of the usefulness of features extracted from unsupervised methods is proposed. The usefulness of these features will be studied on the task of performing named entity recognition within one clinical sub-domain as well as on the task of adapting a named entity recognition model to a new clinical sub-domain. Four named entity types, all very relevant for clinical information extraction, will be studied: Disorder, Finding, Pharmaceutical Drug and Body Structure. The named entity recognition will be performed using conditional random fields. As unsupervised features, a clustering of the semantic representation of words obtained from a random indexing word space will be used.

1 Introduction

Creating the annotated corpus needed for training a NER (named entity recognition) model is costly. This is particularly the case for texts in specialised domains, for which expert annotators are often required. In addition, the need for expert annotators also limits the possibilities of using crowdsourcing approaches (e.g. Amazon Mechanical Turk). Features from unsupervised machine-learning methods, for which no labelled training data is required, have, however, been shown to improve the performance of NER systems (Jonnalagadda et al., 2012). It is therefore likely that by incorporating features from unsupervised methods, it is possible to reduce the amount of training data needed to achieve a fixed level of performance.

Due to differences in the use of language, an NLP system developed for, or trained on, text from one sub-domain often shows a drop in performance when applied on texts from another sub-domain (Martinez et al., 2013). This has the effect that when performing NER on a new sub-domain, annotated text from this new targeted sub-domain might be required, even when there are annotated corpora from other domains. It would, however, be preferable to be able to apply a NER model trained on text from one sub-domain on another sub-domain, with only a minimum of additional data from this other targeted sub-domain. Incorporating features from unsupervised methods might limit the amount of additional annotated data needed for adapting a NER model to a new sub-domain.

The proposed study aims at investigating the usefulness of unsupervised features, both for NER within one sub-domain and for domain adaptation of a NER model. The study has two hypotheses.

- Within one subdomain:

  For reaching the same level of performance when training a NER model, less training data is required when unsupervised features are used.

- For adapting a model trained on one subdomain to a new targeted subdomain:

  For reaching the same level of performance when adapting a NER model to a new subdomain, less additional training data is required in the new targeted subdomain when unsupervised features are used.

For both hypotheses, the level of performance is defined in terms of F-score.

The proposed study will be carried out on different sub-domains within the specialised text domain of *clinical text*.

2 Related research

There are a number of previous studies on named entity recognition in clinical text. For instance, a corpus annotated for the entities Condition,

Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 21–30,
Gothenburg, Sweden, April 26-30 2014. ©2014 Association for Computational Linguistics

Drug/Device and Locus was used for training a support vector machine with uneven margins (Roberts et al., 2008) and a corpus annotated for the entities Finding, Substance and Body was used for training a conditional random fields (CRF) system (Wang, 2009) as well as for training an ensemble of different classifiers (Wang and Patrick, 2009). Most studies have, however, been conducted on the *i2b2 medication challenge* corpus and the *i2b2 challenge on concepts, assertions, and relations* corpus. Conditional random fields (Patrick and Li, 2010) as well as an ensemble classifier (Doan et al., 2012) has for instance been used for extracting the entity Medication names from the *medication challenge* corpus, while all but the best among the top-performing systems used CRF for extracting the entities Medical Problem, Test and Treatment from the *i2b2 challenge on concepts, assertions, and relations* corpus (Uzuner et al., 2011). The best system (de Bruijn et al., 2011) used semi-Markov HMM, and in addition to the features used by most of the other systems (e.g. tokens/lemmas/stems, orthographics, affixes, part-of-speech, output of terminology matching), this system also used features extracted from hierarchical word clusters on un-annotated text. For constructing the clusters, they used Brown clustering, and represented the feature as a 7-bit showing to what cluster a word belonged.

Outside of the biomedical domain, there are many studies on English corpora, which have shown that using features extracted from clusters constructed on unlabelled corpora improves performance of NER models, especially when using a smaller amount of training data (Miller et al., 2004; Freitag, 2004). This approach has also been shown to be successful for named entity recognition in other languages, e.g. German, Dutch and Spanish (Täckström et al., 2012), as well as on related NLP tasks (Biemann et al., 2007), and there are NER tools that automatically incorporate features extracted from unsupervised methods (Stanford, 2012). There are a number of additional studies within the biomedical domain, e.g. using features from Brown and other clustering approaches (Stenetorp et al., 2012) or from k-means clustered vectors from a neural networks-based word space implementation (Pyysalo et al., 2014). Jonnalagadda et al. (2012) also present a study in which unsupervised features are used for training a model on the *i2b2 challenge on concepts, assertions, and relations* corpus. As un-annotated corpus, they used a corpus created by extracting Medline abstracts that are indexed with the publication type ”clinical trials”. They then built a semantic representation of this corpus in the form of a random indexing-based word space. This representation was then used for extracting a number of similar words to each word in the *i2b2 challenge on concepts, assertions, and relations* corpus, which were used as features when training a CRF system. The parameters of the random indexing model were selected by letting the nearest neighbours of a word vote for one of the UMLS categories Medical Problem, Treatment and Test according to the category of the neighbour, and by comparing the category winning the vote to the actual category of the word. The authors motivate their choice of using random indexing for creating features with that this method is scalable to very large corpora without requiring large computational resources.

The method proposed here is similar to the method used by Jonnalagadda et al. (2012). However, the focus of the proposed study is to explore to what extent unsupervised features can help a machine learning system trained only on very little data. It is therefore not feasible to use the large number of features that would be generated by using neighbouring words, as that would require a large training data set to ensure that there are enough training examples for each generated feature. Therefore, the proposed method instead further processes the word space model by constructing clusters of semantically related words, thereby reducing the number of generated features, similar to the approach by Pyysalo et al. (2014).

3 Materials and previous results

Texts from three different clinical sub-domains: *cardiac ICU* (intensive care unit), *orthopaedic ER* (emergency room), and *internal medicine ER* have been annotated (Tables 1-3).[1] All texts are written in Swedish, and they all share the characteristics of text types written under time pressure; all of them containing many abbreviations and incomplete sentences. There are, however, also differences in e.g. what abbreviations are used and what

[1] Research on these texts aiming at extracting information related to Disorders/Findings and Pharmaceutical Drugs has been approved by the Regional Ethical Review Board in Stockholm (Etikprövningsnämnden i Stockholm), permission number 2012/834-31/5.

Data set:	All	
Entity category	# entities	(Unique)
Disorder	1088	(533)
Finding	1798	(1295)
Pharmaceuticals	1048	(497)
Body structure	461	(252)

Table 1: Annotated data, Cardiac ICU

Data set:	All	
Entity category	# entities	(Unique)
Disorder	1258	(541)
Finding	1439	(785)
Pharmaceuticals	880	(212)
Body structure	1324	(423)

Table 2: Annotated data, Orthopaedic ER

entities that are frequently mentioned.

The texts from cardiac ICU and orthopaedic ER will be treated as existing annotations in a current domain, whereas internal medicine ER will be treated as the new target domain. Approximately a third of the texts from internal medicine ER have been doubly annotated, and an evaluation set has been created by manually resolving differences between the two annotators (Skeppstedt et al., 2014). This evaluation subset will be used as held-out data for evaluating the NER task.

The following four entity categories have been annotated (Skeppstedt et al., 2014): (1) Disorder (a disease or abnormal condition that is not momentary and that has an underlying pathological process), (2) Finding (a symptom reported by the patient, an observation made by the physician or the result of a medical examination of the patient), (3) Pharmaceutical Drug (not limited to generic name or trade name, but includes also e.g. drugs expressed by their effect, such as painkiller or sleeping pill). (4) Body Structure (an anatomically defined body part).

These three annotated corpora will be used in the proposed study, together with a large corpus of un-annotated text from which unsupervised features will be extracted. This large corpus will be a subset of the Stockholm EPR corpus (Dalianis et al., 2009), which is a large corpus of clinical text written in Swedish.

Named entity recognition on the internal medicine ER part of the annotated corpus has already been studied, and results on the evaluation set were an F-score of 0.81 for the entity Disorder, 0.69 for Finding, 0.88 for Pharmaceutical Drug, 0.85 for Body Structure and 0.78 for the combined category Disorder + Finding (Skeppstedt et al., 2014). Features used for training the model on the development/training part of the internal medicine ER corpus were the lemma forms of the words, their part of speech, their semantic category in used vocabulary lists, their word constituents (if the words were compounds) as well as the orthographics of the words. A narrow context window was used, as shown by the entries marked in boldface in Figure 1. As terminologies, the Swedish versions of SNOMED CT[2], MeSH[3], ICD-10[4], the Swedish medical list FASS[5] were used, as well as a vocabulary list of non-medical words, compiled from the Swedish Parole corpus (Gellerstam et al., 2000).

4 Methodological background

The proposed method consists of using the training data first for parameter setting (through n-fold cross-validation) and thereafter for training a model using the best parameters. This model is then to be evaluated on held-out data. A number of rounds with parameter setting and training will be carried out, where each new round will make use of an increasingly larger subset of the training data. Two versions of parameter setting and model training will be carried out for each round; one using features obtained from unsupervised methods on un-annotated text and one in which such features are not used. The results of the two versions are then to be compared, with the hypothesis that the model incorporating unsupervised methods will perform better, at least for small training data sizes.

To accomplish this, the proposed method makes use of four main components: (1) A system for training a NER model given features extracted from an annotated corpus. As this component, a conditional random fields (CRF) system will be used. (2) A system for automatic parameter setting. As a large number of models are to be constructed on different sizes of the training data, for which optimal parameters are likely to differ, parameters for each set of training data has to be determined automatically for it to be feasible to

[2] www.ihtsdo.org

[3] mesh.kib.ki.se

[4] www.who.int/classifications/icd/en/

[5] www.fass.se

Data set:	Development		Final evaluation
Entity category	# entities	(Unique)	# entities
Disorder	1,317	(607)	681
Finding	2,540	(1,353)	1282
Pharmaceuticals	959	(350)	580
Body structure	497	(197)	253
Tokens in corpus	45,482		25,370

Table 3: Annotated entities, internal medicine ER

Token	*Lemma*	*POS*	*Terminology*	*Compound*		*Orthographics*	*Cluster membership level 1*	..	*Cluster membership level n*	*Category*	
DVT	*dvt*	*noun*	*disorder*	-	-	*all upper*	*#40*	..	*#39423*	*B-Disorder*	
patient	*patient*	**noun**	*person*	-	-	-	*#3*	..	*#23498*	*O*	
with	**with**	**prep.**	**parole**	-	-	-	*#14*	..	*#30892*	*O*	
chestpain	**chestpain**	**noun**	**finding**	**chest**	**pain**	**-**	*#40*	..	*#23409*	*B-Finding*	← **Current**
and	*and*	**conj.**	*parole*	-	-	-	-	..	-	*O*	
problems	*problem*	*noun*	*finding*	-	-	-	*#40*	..	*#23409*	*B-Finding*	
to	*to*	*prep.*	*finding*	-	-	-	-	..	-	*I-Finding*	
breathe	*breathe*	*verb*	*finding*	-	-	-	*#90*	..	*#23409*	*I-Finding*	

Figure 1: A hypothetical example sentence, with hypothetical features for training a machine learning model. Features used in a previous medical named entity recognition study (Skeppstedt et al., 2014) on this corpus are shown in boldface. The last column contains the entity category according to the manual annotation.

carry out the experiments. (3) A system for representing semantic similarity of the words in the un-annotated corpus. As this component, a random indexing based word space model will used. (4) A system for turning the semantic representation of the word space model into features to use for the NER model. As this component, clustering will be used.

To give a methodological background, the theoretical foundation for the four components will be described.

4.1 Conditional random fields

Conditional random fields (CRF or CRFs), introduced by Lafferty et al. (2001), is a machine learning method suitable for segmenting and labelling sequential data and therefore often used for e.g. named entity recognition. As described in the related research section, CRFs have been used in a number of studies for extracting entities from clinical text. In contrast to many other types of data, observed data points for sequential data, such as text, are dependent on other observed data points. Such dependences between data points are practical to describe within the framework of graphical models (Bishop, 2006, p. 359), to which CRF belongs (Sutton and McCallum, 2006, p. 1). In the special, but frequently used, case of linear chain CRF, the output variables are linked in a chain. Apart from being dependent on the input variables, each output variable is then conditionally independent on all other output variables, except on the previous and following output variable, given these two neighbouring output variables. In a named entity recognition task, the output variables are the named entity classes that are to be predicted and the observed input variables are observed features of the text, such as the tokens or their part-of-speech.

CRF is closely related to Hidden Markov Models, which is also typically described as a graphical model. A difference, however, is that Hidden Markov Models belongs to the class of generative models, whereas CRF is a conditional model (Sutton and McCallum, 2006, p. 1). Generative models model the joint distribution between input variables and the variables that are to be predicted (Bishop, 2006, p. 43). In contrast, CRF and other conditional models instead directly model the conditional distribution, enabling the use of a larger feature set (Sutton and McCallum, 2006, p. 1).

For named entity recognition, the IOB-encoding is typically used for encoding the output variables. Tokens not annotated as an entity are then encoded with the label *O*, whereas labels for annotated tokens are prefixed with a *B*, if it is the first token in the annotated chunk, and an *I* otherwise (Jurafsky and Martin, 2008, pp. 763–764). An example of this encoding is shown in the last column in Figure 1. In this case, where there are four types of entities, the model thus learns to classify in 8+1 different classes: B-Disorder, I-Disorder, B-Finding, I-Finding, B-Drug, I-Drug, B-BodyStructure, I-BodyStructure and *O*.

The dependencies are defined by a large number of (typically binary) feature functions of input and output variables. E.g. is all of the following true?

- Output: The output at the current position is **I-Disorder**
- Output: The output at the previous position is **B-Disorder**
- Input: The token at the current position is **chest-pain**
- Input: The token at the previous position is **experiences**

A feature function in a linear chain CRF can only include the values of the output variable in current position and in the immediate previous position, whereas it can include, and thereby show a dependence on, input variables from any position.

The CRF model is trained through setting weights for the feature functions, which is carried out by penalised maximum likelihood. *Penalised* means that regularisation is used, and regularisation is performed by adding a penalty term, which prevents the weights from reaching too large values, and thereby prevents over-fitting (Bishop, 2006, p. 10). The L1-norm and the L2-norm are frequently used for regularisation (Tsuruoka et al., 2009), and a variable C governs the importance of the regularisation. Using the L1-norm also results in that if C is large enough, some of the weights are driven to zero, resulting in a sparse model and thereby the feature functions that those weights control will not play any role in the model. Thereby, complex models can be trained also on data sets with a limited size, without being over-fitted. However, a suitable value of C must still be determined (Bishop, 2006, p. 145).

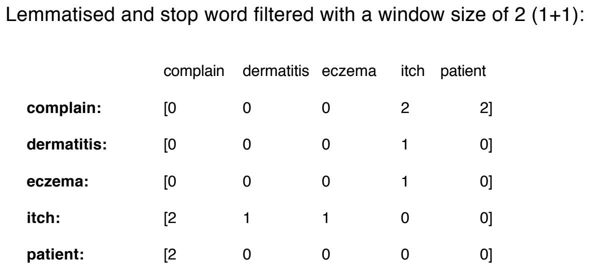
The plan for the proposed study is to use the CRF package CRF++[6], which has been used in a number of previous NER studies, also in the medical domain. The CRF++ package automatically generates feature functions from user-defined templates. When using CRF++ as a linear chain CRF, it generates one binary feature function for each combination of output class, previous output class and unique string in the training data that is expanded by a template. This means that L * L * M feature functions are generated for each template, where L = the number of output classes and M = the number of unique expanded strings. If only the current token were to be used as a feature, the number of feature functions would be $9 * 9 * |unique\ tokens\ in\ the\ corpus|$. In practice, a lot of other features are, however, used. Most of these features will be of no use to the classifier, which means that it is important to use an inference method that sets the weights of the feature functions with irrelevant features to zero, thus an inference method that promotes sparsity.

4.2 Parameter setting

As previously explained, a large number of models are to be constructed, which requires a simple and efficient method for parameter setting. An advantage with using the L1-norm is that only one parameter, the C-value, has to be optimised, as the weights for feature functions are driven to zero for feature functions that are not useful. The L1-norm will therefore be used in the proposed study. A very large feature set can then be used, without running the risk of over-fitting the model. Features will include those that have been used in previous clinical NER studies (Jonnalagadda et al., 2012; de Bruijn et al., 2011; Skeppstedt et al., 2014), with a context window of four previous and four following tokens.

When maximising the conditional log likelihood of the parameters, the CRF++ program will set parameters that are optimal for training the model for the best micro-averaged results for the four classes Disorder, Finding, Pharmaceutical drug and Body structure. A hill climbing search (Marsland, 2009, pp. 262–264) for finding a good C-value will be used, starting with a value very close to zero and thereafter changing it in a direction that improves the NER results. A decreasingly smaller step size will be used for changing

[6] crfpp.sourceforge.net



Lemmatised and stop word filtered with a window size of 2 (1+1):

	complain	dermatitis	eczema	itch	patient
complain:	[0	0	0	2	2]
dermatitis:	[0	0	0	1	0]
eczema:	[0	0	0	1	0]
itch:	[2	1	1	0	0]
patient:	[2	0	0	0	0]

Figure 2: Term-by-term co-occurrence matrix for the small corpus "Patient complains of itching dermatitis. Patient complains of itching eczema."

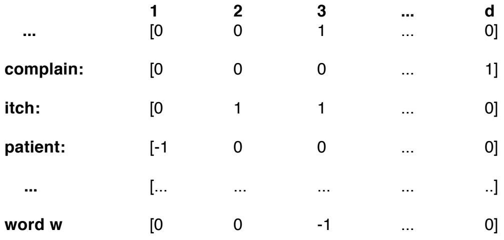
the C-value, until only small changes in the results can be observed.

4.3 Random indexing

Random indexing is one version of the word space model, and as all word space models it is a method for representing distributional semantics. The random indexing method was originally devised by Kanerva et al. (2000), to deal with the performance problems (in terms of memory and computation time) that were associated with the LSA/LSI implementations at that time. Due to its computational efficiency, random indexing remains to be a popular method when building distributional semantics models on very large corpora, e.g. large web corpora (Sahlgren and Karlgren, 2009) or Medline abstracts (Jonnalagadda et al., 2012).

Distributional semantics is built on the distributional hypothesis, which states that "Words with similar meanings tend to occur in similar contexts". If *dermatitis* and *eczema* often occur in similar contexts, e.g. "Patient complains of itching *dermatitis*" and "Patient complains of itching *eczema*", it is likely that *dermatitis* and *eczema* have a similar meaning. One possible method of representing word co-occurrence information is to construct a term-by-term co-occurrence matrix, i.e. a matrix of dimensionality $w \times w$, in which w is the number of terms (unique semantic units, e.g. words) in the corpus. The elements of the matrix then contain the number of times each semantic unit occurs in the context of each other semantic unit (figure 2).

The *context vectors* of two semantic units can then be compared as a measure of semantic similarity between units, e.g. using the the euclidian distance between normalised context vectors or the cosine similarity.



	1	2	3	...	d
...	[0	0	1	...	0]
complain:	[0	0	0	...	1]
itch:	[0	1	1	...	0]
patient:	[-1	0	0	...	0]
...	[...	...	...	...	..]
word w	[0	0	-1	...	0]

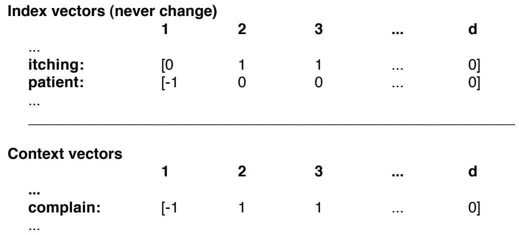
Figure 3: Index vectors.

The large dimension of a term-by-term matrix leads, however, to scalability problems, and the typical solution to this is to apply dimensionality reduction on the matrix. In a semantic space created by latent semantic analysis, for instance, dimensionality reduction is performed by applying the linear algebra matrix operation singular value decomposition (Landauer and Dutnais, 1997). *Random indexing* is another solution, in which a matrix with a smaller dimension is created from start, using the following method (Sahlgren et al., 2008):

Each term in the data is assigned a unique representation, called an *index vector*. The index vectors all have the dimensionality d (where $d \geq 1000$ but $\ll w$). Most of the elements of the index vectors are set to 0, but a few, randomly selected, elements are set to either +1 or -1. (Usually around 1-2% of the elements.) Instead of having orthogonal vectors, as is the case for the term-by-term matrix, the index vectors are nearly orthogonal. (See Figure 3.)

Each term in the data is also assigned a *context vector*, also of the dimensionality d. Initially, all elements in the context vectors are set to 0. The context vector of each term is then updated by, for every occurrence of the term in the corpus, adding the index vectors of the neighboring words. The neighboring words are called the *context window*, and this can be both narrow or wide, depending on what semantic relations the word space model is intended to capture. The size of the context window can have large impact on the results (Sahlgren et al., 2008), and for detecting paradigmatic relations (i.e. words that occur in similar contexts, rather than words that occur together) a fairly narrow context window has been shown to be most effective.

The resulting context vectors form a matrix of dimension $w \times d$. This matrix is an approximation of the term-by-term matrix, and the same similar-



Index vectors (never change)

	1	2	3	...	d
...					
itching:	[0	1	1	...	0]
patient:	[-1	0	0	...	0]
...					

Context vectors

	1	2	3	...	d
...					
complain:	[-1	1	1	...	0]
...					

Figure 4: The updated context vectors.

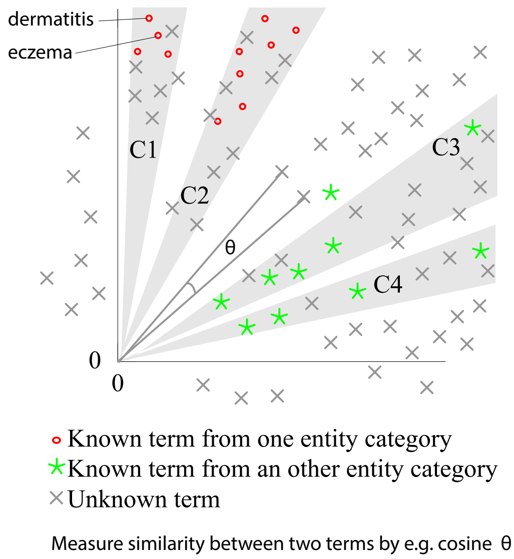

dermatitis
eczema
C1
C2
C3
C4
θ
0
0
○ Known term from one entity category
✳ Known term from an other entity category
× Unknown term
Measure similarity between two terms by e.g. cosine θ

Figure 5: Context vectors for terms in a hypothetical word space with *d*=2. The context vectors for the semantically similar words *eczema* and *dermatitis* are close in the word space, in which closeness is measured as the cosine of the angle between the vectors. Four hypothetical clusters (C1-C4) of context vectors are also shown; clusters that contain a large proportion of known terms.

ity measures can be applied.

A hypothetical word space with *d*=2 is shown in Figure 5.

4.4 Clustering

As mentioned earlier, for the word space information to be useful for training a CRF model on a small data set, it must be represented as a feature that can only take a limited number of different values. The proposed methods for achieving this is to cluster the context vectors of the word space model, similar to what has been done in previous research (Pyysalo et al., 2014). Also similar to previous research, cluster membership for a word in the NER training and test data will be used as a feature. Four named hypothetical clusters of context vectors are shown in the word space model in Figure 5 to illustrate the general idea, and an example of how to use cluster membership as a feature is shown Figure 1.

Different clustering techniques will be evaluated, for the quality of the created clusters, as well as for their computational efficiency. Having hierarchical clusters might be preferable, as cluster membership to clusters of different granularity then can be offered as features for training the CRF model. Which granularity that is most suitable might vary depending on the entity type and also depending on the size of the training data. However, e.g. performing hierarchical agglomerative clustering (Jurafsky and Martin, 2008, p. 700) on the entire unlabelled corpus might be computationally intractable (thereby defeating the purpose of using random indexing), as it requires pairwise comparisons between the words in the corpus. The pairwise comparison is a part of the agglomerative clustering algorithm, in which each word is first assigned its own cluster and then each pair of clusters is compared for similarity, resulting in a merge of the most similar clusters. This process is thereafter iteratively repeated, having the distance between the centroids of the clusters as similarity measure. An alternative, which requires a less efficient clustering algorithm, would be to not create clusters of all the words in the corpus, but to limit initially created clusters to include those words that occur in available terminologies. Cluster membership of unknown words in the corpus could then be determined by measuring similarity to the centroids of these initially created clusters.

Regardless of what clustering technique that is chosen, the parameters of the random indexing models, as well as of the clustering, will be determined by evaluating to what extent words that belong to one of the studied semantic categories (according to available terminologies) are clustered together. This will be measured using *purity* and *inverse purity* (Amigó et al., 2009). However, if clusters are to be created from all words in the corpus, the true semantic category will only be known for a very small subset of clustered words. In that case, the two measures have to be defined as *purity* being to what extent a cluster only contains known words of one category and *inverse purity* being the extent to which known words of the same category are grouped into the same cluster.

5 Proposed experiments

The first phase of the experiments will consist of finding the best parameters for the random indexing model and the clustering, as described above.

The second phase will consist of evaluating the usefulness of the clustered data for the NER task. Three main experiments will be carried out in this phase (I, II and III), using data set(s) from the following sources:

I: Internal medicine ER

II: Internal medicine ER + Cardiac ICU

III: Internal medicine ER + Orthopaedic ER

In each experiment, the following will be carried out:

1. Divide internal medicine ER training data into 5 partitions (into a random division, to better simulate the situation when not all data is available, using the same random division for all experiments).
2. Run step 3-5 in 5 rounds. Each new round uses one additional internal medicine ER partition: (Experiments II and III always use the entire data set from the other domain). In each round, two versions of step 3-5 will be carried out:
   (a) With unsupervised features.
   (b) Without unsupervised features.
3. Use training data for determining C-value (by n-fold cross-validation).
4. Use training data for training a model with this C-value.
5. Evaluate the model on the held-out internal medicine ICU data.

6 Open issues

What clustering technique to use has previously been mentioned as one important open issue. The following are examples of other open issues:

- Could the information obtained from random indexing be used in some other way than as transformed to cluster membership features? Jonnalagadda et al. (2012) used the terms closest in the semantic space as a feature. Could this method be adapted in some way to models constructed with a small amount of training data? For instance by restricting what terms are allowed to be used as such a feature, and thereby limiting the number of possible values this feature can take.
- Would it be better to use other approaches (or compare different approaches) for obtaining features from unlabelled data? A possibility could be to use a more standard clustering approach, such as Brown clustering used in previous clinical NER studies (de Bruijn et al., 2011). Another possibility could be to keep the idea of creating clusters from vectors in a word space model, but to use other methods than random indexing for constructing the word space; e.g. the previously mentioned latent semantic analysis (Landauer and Dutnais, 1997), or a neural networks-based word space implementation (Pyysalo et al., 2014).
- Many relevant terms within the medical domain are multi-word terms (e.g. of the type *diabetes mellitus*), and there are studies on how to construct semantic spaces with such multiword terms as the smallest semantic unit (Henriksson et al., 2013). Should the whitespace segmented token be treated as the smallest semantic unit in the proposed study, or should the use of larger semantic units be considered?

Acknowledgements

Many thanks to Aron Henriksson, Alyaa Alfalahi, Maria Kvist, Gunnar Nilsson and Hercules Dalianis for taking a very active part in the planing of the proposed study, as well as to the three anonymous reviewers for their constructive and detailed comments.

References

Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Inf. Retr.*, 12(4):461–486, aug.

Chris Biemann, Claudio Giuliano, and Alfio Gliozzo. 2007. Unsupervised part of speech tagging supporting supervised methods. In *RANLP*.

Christopher M. Bishop. 2006. *Pattern recognition and machine learning*. Springer, New York, NY.

Hercules Dalianis, Martin Hassel, and Sumithra Velupillai. 2009. The Stockholm EPR Corpus - Characteristics and Some Initial Findings. In *Proceedings of ISHIMR 2009, Evaluation and implementation of e-health and health information initiatives: international perspectives. 14th International Symposium for Health Information Management Research, Kalmar, Sweden*, pages 243–249.

Berry de Bruijn, Colin Cherry, Svetlana Kiritchenko, Joel D. Martin, and Xiaodan Zhu. 2011. Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010. *J Am Med Inform Assoc*, 18(5):557–562.

Son Doan, Nigel Collier, Hua Xu, Hoang Duy Pham, and Minh Phuong Tu. 2012. Recognition of medication information from discharge summaries using ensembles of classifiers. *BMC Med Inform Decis Mak*, 12:36.

Dayne Freitag. 2004. Trained named entity recognition using distributional clusters. In *EMNLP*, pages 262–269.

M Gellerstam, Y Cederholm, and T Rasmark. 2000. The bank of Swedish. In *LREC 2000. The 2nd International Conference on Language Resources and Evaluation*, pages 329–333, Athens, Greece.

Aron Henriksson, Mike Conway, Martin Duneld, and Wendy W. Chapman. 2013. Identifying synonymy between SNOMED clinical terms of varying length using distributional analysis of electronic health records. In *Proceedings of the Annual Symposium of the American Medical Informatics Association (AMIA 2013)*, Washington DC, USA.

Siddhartha Jonnalagadda, Trevor Cohen, Stephen Wu, and Graciela Gonzalez. 2012. Enhancing clinical concept extraction with distributional semantics. *J Biomed Inform*, 45(1):129–40, Feb.

Daniel Jurafsky and James H. Martin. 2008. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall, second edition, February.

Pentti Kanerva, Jan Kristoferson, and Anders Holst. 2000. Random indexing of text samples for latent semantic analysis. In L. R. Gleitman and A. K. Joshi, editors, *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, Mahwah, NJ.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA.

Thomas K Landauer and Susan T. Dutnais. 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, pages 211–240.

Stephen Marsland. 2009. *Machine learning : an algorithmic perspective*. Chapman & Hall/CRC, Boca Raton, FL.

David Martinez, Lawrence Cavedon, and Graham Pitson. 2013. Stability of text mining techniques for identifying cancer staging. In *Proceedings of the 4th International Louhi Workshop on Health Document Text Mining and Information Analysis - Louhi 2013*, Sydney, Australia, February.

Scott Miller, Jethran Guinness, and Alex Zamanian. 2004. Name tagging with word clusters and discriminative training. In *Proceedings of HLT*, pages 337–342.

Jon Patrick and Min Li. 2010. High accuracy information extraction of medication information from clinical notes: 2009 i2b2 medication extraction challenge. *J Am Med Inform Assoc*, 17(5):524–527, Sep-Oct.

Sampo Pyysalo, Filip Ginter, Hans Moen, Tapio Salakoski, and Sophia Ananiadou. 2014. Distributional semantics resources for biomedical text processing. In *Proceedings of Languages in Biology and Medicine*.

Angus Roberts, Robert Gaizasukas, Mark Hepple, and Yikun Guo. 2008. Combining terminology resources and statistical methods for entity recognition: an evaluation. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pages 2974–2979, Marrakech, Morocco, may. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.

Magnus Sahlgren and Jussi Karlgren. 2009. Terminology mining in social media. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09.

Magnus Sahlgren, Anders Holst, and Pentti Kanerva. 2008. Permutations as a means to encode order in word space. In *Proceedings of the 30th Annual Meeting of the Cognitive Science Society*, pages 1300–1305.

Maria Skeppstedt, Maria Kvist, Gunnar H Nilsson, and Hercules Dalianis. 2014. Automatic recognition of disorders, findings, pharmaceuticals and body structures from clinical text: An annotation and machine learning study. *J Biomed Inform*, Feb (in press).

NLP Group Stanford. 2012. Stanford Named Entity Recognizer (NER). http://www-nlp.stanford.edu/software/CRF-NER.shtml. Accessed 2012-03-29.

Pontus Stenetorp, Hubert Soyer, Sampo Pyysalo, Sophia Ananiadou, and Takashi Chikayama. 2012. Size (and domain) matters: Evaluating semantic word space representations for biomedical text. In *Proceedings of the 5th International Symposium on Semantic Mining in Biomedicine*.

Charles. Sutton and Andrew McCallum. 2006. An introduction to conditional random fields for relational learning. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual Word Clusters for Direct Transfer of Linguistic Structure. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 477–487, Montréal, Canada, June. Association for Computational Linguistics.

Yoshimasa Tsuruoka, Jun'ichi Tsujii, and Sophia Ananiadou. 2009. Fast full parsing by linear-chain conditional random fields. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 790–798, Stroudsburg, PA, USA. Association for Computational Linguistics.

Özlem. Uzuner, Brett R. South, Shuying Shen, and Scott L. DuVall. 2011. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *J Am Med Inform Assoc*, 18(5):552–556.

Yefeng Wang and Jon Patrick. 2009. Cascading classifiers for named entity recognition in clinical notes. In *Proceedings of the Workshop on Biomedical Information Extraction*, pages 42–49.

Yefeng Wang. 2009. Annotating and recognising named entities in clinical notes. In *Proceedings of the ACL-IJCNLP Student Research Workshop*, pages 18–26, Singapore.

Now We Stronger Than Ever: African-American Syntax in Twitter

Ian Stewart
Dartmouth College
Hanover, NH 03755
ian.b.stewart.14@dartmouth.edu

Abstract

African American English (AAE) is a well-established dialect that exhibits a distinctive syntax, including constructions like habitual *be*. Using data mined from the social media service Twitter, the proposed senior thesis project intends to study the demographic distribution of a subset of AAE syntactic constructions. This study expands on previous sociolinguistic Twitter work (Eisenstein et al., 2011) by adding part-of-speech tags to the data, thus enabling detection of short-range syntactic features. Through an analysis of ethnic and gender data associated with AAE tweets, this project will provide a more accurate description of the dialect's speakers and distribution.

1 Introduction

Most modern studies of sociolinguistics focus on phonetic or lexical variation to draw conclusions about a dialect or a social group. For example, the Atlas of North American English (2005) maps language variation entirely by the differences in production and perception of phonetic variables. Although this is an integral part of sociolinguistics, research has given less attention to synchronic variation in syntax, which is also an important aspect of language change. Recent initiatives like Yale's Grammatical Diversity Project (2014) have been invaluable in demonstrating the breadth of syntactic variation in North America, and smaller-scale research like Kendall et al. (2011) has been equally vital for investigating the properties of constructions within a "nonstandard" dialect. While other sociolinguistic studies have used a systematic analysis of corpora to detect phonetic and lexical change (Yaeger-Dror and Thomas, 2010; Eisenstein et al., 2011), such approaches are under-utilized with respect to syntactic variation.

Varieties of African American English provide a wide range of syntactic features to study, with constructions ranging from aspectual particles like *done* (such as "he done eaten" for "he's just eaten") to double negation (such as "can't nobody") (Wolfram, 2004). AAE shares some features with Southern American English but is spoken throughout the United States. The majority of research in AAE syntax relies on data collected from interview-based conversations (Labov, 2012), published letters (Kendall et al., 2011) and observations of dialect acquisition in children (Green and Roeper, 2007). Though valuable, this kind of data is often restricted to a specific location and cannot always keep pace with the most recent language developments among fluent young speakers. The proposed study seeks to systematically study AAE syntax in a more youth-centric environment and describe the geographical or gender-based correlation in the distribution of such syntax.

2 Proposal

This thesis's primary hypothesis is that there is a quantifiable correlation between ethnicity and features of AAE syntax found in large-scale social media. This will be supported or challenged by the geographic and demographic data associated with the constructions, as previous studies of dialect reappropriation have suggested a spread of AAE beyond expected areas (Reyes, 2005). As a secondary hypothesis, the project will investigate a correlation between AAE syntax and gender, which has been suggested but not tested on a large scale. Eckert and McConnell-Ginet (2013) argue for a connection between gender and identity expression (often associated with "speech style"), which would generally suggest greater AAE syntax usage among women. Even if the neither correlation is proven plausible, the study will provide valuable insight about the frequency and ge-

Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 31–37,
Gothenburg, Sweden, April 26-30 2014. ©2014 Association for Computational Linguistics

ographic location of specific AAE syntactic features. This project is being co-supervised by a professor of sociolinguistics and a postdoctoral researcher in computer science.

3 Procedure

3.1 Preprocessing

As a data source, the online social media service Twitter is a firehose of information, comprising 16% of all Internet users (Duggan and Brenner, 2013) and millions of "tweets" (140-character posts) per day. Using data from Twitter, Eisenstein et al. (2011) demonstrated an empirical correlation between regional vocabulary and the location of Twitter users. In a similar approach, this project combines metadata of tweets with their content and uses this information to investigate the relationship between AAE syntax and region.

The Twitter data was collected from July to December 2013. We used the website's API that provides a stream of publicly available tweets (approximately 5% of the total tweet volume), restricting our data to geotagged tweets from within the United States. Each tweet includes geographical coordinates (latitude and longitude), name and identity of the Twitter user, and time of creation, as well as its content. The content is broken up and simplified in separate tokens for analysis (e.g. "What's up?" becomes "[what] [' s] [up] [?]"). Following previous work (Eisenstein et al., 2010), we minimize spam posts by removing tweets that contain URLs, and tweets from users that contributed fewer than 20 messages to this data. This gives us a corpus of about 200 million tweets.

Before mining the data, we seek to first eliminate as many retweets as possible to avoid skewing the data. Although we can easily detect retweets that are made through the standard Twitter interface, or are preceded by the token *RT*, we notice that the data contains several unstructured retweets, where a user quotes a tweet from another user without explicitly indicating that it is a retweet. We handle these by simply filtering out every line containing a high-frequency higher order n-gram. After qualitatively observing the results of filtering with different n-gram and frequency combinations, the most efficient and least error-prone filter was determined to be a 6-gram with frequency over 10. Making the assumption that most retweets occur within the same 24-hour period, the tweets of each day were segmented into 6-grams. The 6-grams were tabulated, and all tweets containing a 6-gram with frequency over 10 were omitted. Each day's filtered tweets were then recombined to form the full monthly data. This reduced the size of the corpus by about 26%.

After being filtered, the content of each tweet is fed into a part-of-speech (POS) tagging program developed by Gimpel et al. (2011). This program has achieved over 90% accuracy by using statistics gathered from Twitter data hand-labeled with POS tags. The tagging task is accomplished with a conditional random field using features including non-standard orthography, distributional similarity, and phonetic normalization.

The above uses only 25 tags that range from simple lexemes like O (non-possessive pronoun) to complex morphemes like M (proper noun + verbal). In addition to these basic POS tags, the tweets were tagged with a Penn Treebank-style model trained over another hand-labelled data set (Derczynski et al., 2013). This additional tag set is crucial in detecting constructions like 3rd-person singular -s drop (e.g. "she keep her face down"), which depends on verbal morphology that can be described with PTB tags, but not the simplified tagset of Gimpel et al. (2011).

Owoputi et al. (2013) address the possibility that some AAE tense-aspect-mood (TAM) particles may fall outside the standard POS-tag systems. However, we have observed that "nonstandard" morphemes like *finna* were tagged similarly to Standard American English morphemes, which is likely due to the AAE morphemes exhibiting similar distributional properties to corresponding standard morphemes.

3.2 Querying and Analysis

Using the preprocessed data, it is possible to search through the tagged tweets for a particular syntactic construction by combining the lexical and POS information in a search phrase. For instance, one might use the phrase PRO-ADJ ("we cool," "he cute") to detect copula deletion or PRO-be-V for habitual be. Using regular expressions, these searches can be fine-tuned to ignore noise in the data by searching for patterns like !V-PRO-ADJ ("non-verb+pronoun+adjective"), which ignore false positives like "made me hot." In addition, cases of long-distance constructions like negative concord ("there ain't nobody") can be handled by

Table 1: AAE Constructions and Patterns of Detection

Construction	Example from Corpus	Simplified Pattern	Tagger Used
copula deletion	we stronger than ever	not(V)+PRO+ADJ	PTB
habitual *be*	now i be sober af	not(V)+PRO+*be*+ADJ	PTB
continuative *steady*	steady getting bigger	*steady*+not(N)	Gimpel
completive *done*	u done pissed me off	*done*+V_{PST}	PTB
future *finna (fixing to)*	i'm finna tweet	*finna*+V	Gimpel
remote past *been*	i been had it	PRO/N+*been*+V_{PST}	PTB
negative concord	don't say nothing	*don't/ain't/can't*+V+ *nobody/nothing/nowhere/no*	Gimpel
null genitive marking	time of they life	PRO_{NOM}+N	Gimpel
ass camouflage construction (Collins et al. 2008)	divorced his ass	V+PRO_{POSS}+*ass*	PTB

accounting for a wider context than the keywords themselves, using gaps in the expression. For instance, we detected copula deletion with !V-PRO-ADJ as well as !V-PRO-ADV-ADJ. This strategy was especially useful in preventing false negatives that would otherwise be filtered by rigid patterns (e.g. “he too cute” ignored by !V-PRO-ADJ).

Table 1 contains a list of all constructions queried for this project. To the extent of our knowledge, this is the first study to use regular expressions to use regular expressions and POS tagged data to capture “non-standard” English syntax. The “Tagger” column refers to the POS tagger used to detect the construction: either “Gimpel” (Gimpel et al., 2011) or “PTB” (Derczynski et al., 2013).

Some of the constructions, such as the null genitive (e.g. “time of they life”), could be classified as morphological rather than syntactic phenomena and thus may appear to fall outside the scope of this project. However, it must be noted that these phenomena would not be easily detectable without a POS tagger, which relies on the syntactic context to accurately tag such ambiguous words as “they” (which could be a misspelling of “their”). Furthermore, studies such as Wolfram (2004) that survey AAE grammar also consider morphological phenomena to have comparable frequency and distributional tendencies as syntactic phenomena. Thus, this project chooses to analyze such morphological patterns in the same manner as syntactic patterns.

After querying the data using the regular expressions, the resulting tweets are associated with the metadata corresponding to each tweet. This includes demographic information about the ZIP Code Tabulation Area (ZCTA) associated with the tweet (based on the latitude and longitude coordinates) as well as the estimated gender of the tweeter. ZCTAs are regions defined by the Census Bureau that roughly correspond to postal ZIP codes. Each ZCTA's demographic data includes a number of features. We focus on ethnicity population percentages, overall population in the ZCTA, median age, and percentage of the population living in rented housing (which in some cases could be used to approximate a ZIP code's relative “urban-ness”). The gender of a user is guessed by comparing the tweeter's name with the Social Security Administration's list of baby names from 1995 (`http://www.ssa.gov/oact/babynames/limits.html`), with any user whose name does not appear in the list being assigned a gender of “Unknown”. This is a common method used to determine gender in large-scale datasets (Sloan et al., 2013) and one suited to Twitter's younger user base (Duggan and Brenner, 2013).

4 Results

4.1 Comparison of Average Demographics

Our initial approach to the hypothesis – namely, that Twitter shows a quantifiable correlation between ethnicity and usage of AAE syntax – was a comparison of the demographics of the tweeters that use the AAE constructions listed in Table 1 to the average demographics over all users in our data. The constructions' average demographics

Table 2: Mean Demographic Profiles of AAE Construction Users

Construction	User %	Mean % African-American Population	Mean % Caucasian Population	Gender Ratio Female : Male : Unknown
Overall Statistics	1, 135, 019 users total	13.67 ± 18.66%	71.81 ± 21.66%	36.78 : 31.17 : 32.05
Copula Deletion	45.62%	13.64%	71.80%	37.27 : 30.18 : 32.55
ass Camouflage Construction	40.25%	14.09%	71.4%	36.27 : 28.88 : 34.84
Future *finna*	17.33%	14.46%	70.97%	35.37 : 27.65 : 36.98
Habitual *be*	31.63%	14.43%	71.24%	36.04 : 28.44 : 35.52
Continuative *steady*	1.304%	15.45%	69.44%	33.20 : 26.32 : 40.48
Completive *done*	6.061%	14.81%	70.44%	34.06 : 26.95 : 38.98
Remote Past *been*	8.384%	14.83%	70.48%	33.58 : 25.99 : 36.80
Negative Concord	18.14%	14.47%	70.92%	35.30 : 27.70 : 37.00
Negative Inversion	17.66%	14.50%	70.92%	35.30 : 27.63 : 37.07
Null Genitive	13.59%	14.61%	70.75%	34.84 : 27.56 : 37.60

were calculated counting each construction-user only once, regardless of how many times they use that construction.

While reasonable, this approach did not provide encouraging results, as demonstrated by Table 2. The constructions' demographics deviated only slightly from the overall demographics, though the variation reflected the expected trend of higher African-American population (avg. +0.859%) and lower Caucasian population (avg. -0.974%). The constructions showed similar standard deviations to those of the overall demographics. Further ethnic statistics such as average Asian population, which might have been interesting in light of research on dialect reappropriation (Reyes, 2005), were also highly uniform when comparing constructions to overall data.

In addition to ethnic demographics, the gender breakdown was somewhat uninformative as both female and male users were less represented than expected. This may have indicated a failure on the part of the gender-guesser to guess more unusual names like "Notorious J. $tash" that could be associated with AAE syntax. With such negligible deviations from the mean demographics, additional data analysis techniques such as linear regression and clustering of users with similar demographic data would seem to yield negligible results. Thus, these techniques were deemed unnecessary for these averages.

There are a few possible explanations for the inconclusive results in ethnic demographics and gender. First, the information associated with the ZCTA is drawn from the 2010 U.S. census data, which may not match the demographics of social media users. While the time difference between 2010 and 2013 is unlikely to make a significant difference, the discrepancy between real-life statistics and social media metadata may result in statistics contradictory to the Twitter user demographics proposed by Duggan and Brenner (2013). The current study accepts this as a possible source of error and looks toward future studies that directly associate social media users with geographic demographics. More importantly, this thesis relies on ethnic demographics derived from users' environments rather than directly available data such as names, as in Eisenstein et al. (2011). This distinction is crucial, as it dampens the apparent presence of black Twitter users in ZCTAs with low African American population percentages.

While statistically inconclusive for individual constructions, the apparent pervasiveness of AAE syntax as a whole is surprising, even considering the observation by Duggan and Brenner (2013) that 26% of African-American Internet users are on Twitter. Admittedly, no regular expression is free from error, but the apparent 45.62% copula deletion usage rate is impressive for a construction that was once used to parody the speech of AAE speakers (Green, 2002). Furthermore, the users of each construction tend to be located in the data's most common ZCTAs, which are often youth-centric college towns such as San Marcos, Texas. The non-trivial user percentages and significant diffusion of usage outside of expected urban areas build on claims by Wolfram (2004) about "new and intensifying structures in urban AAVE," such as habitual *be*, as well as "receding urban features" such as remote past *been*. The relatively homogenous distribution of such constructions may even reflect a stable position for AAE as a unified dialect across typical American English dialect regions. However, a long-term Twitter corpus will be necessary to test the diachronic behavior of these apparently " receding" and " intensifying" features.

4.2 Logistic Regression

Following the initial results, we adopted a different approach to measure AAE usage by performing a logistic regression over the demographics collected for the AAE constructions as well as their Standard American English (SAE) counterparts. For example, the SAE equivalent of the AAE future *finna* was considered to be regular genitive pronouns (e.g. AAE " they house" vs. SAE "their house"). At the time of submission, we only extracted SAE demographics for a subset of the constructions. The most salient results of the regression are displayed in Table 3. The variables under consideration are the correlation coefficients relating each construction to the demographics associated with the users, with positive values indicating a trend toward the AAE construction and negative values indicating a trend toward the SAE construction.

Before observing the coefficients, the first notable characteristic of the SAE data is the high rate of occurrence for most standard constructions, such as " Standard *be*+V_{ing}" . This may indicate that there is overlap in SAE and AAE usage among Twitter users, which is unsurprising given the prevalence of code-switching among AAE speakers in non-virtual environments (Labov, 2012) as well as the strong potential for dialect spread (Reyes, 2005). To investigate this possibility, future refinement of this regression approximation will compare Twitter users who only employ SAE constructions versus those who only employ the corresponding AAE construction. Though perhaps an artificial distinction that will tend more toward data sparsity than abundance, this strategy will hopefully reveal a split between speakers that tend more toward one dialect than the other, from which further proposals can be tested (e.g. the most reliable construction characterizing each dialect).

The correlation coefficients in Table 3 generally tend toward positive for population of the ZCTA, suggesting a prevalence of AAE in high-population areas and a diffusion of SAE throughout all populated areas. However, the correlation coefficients for Caucasian population and African-American population are less informative and tend slightly toward SAE constructions, with the notable exceptions of negative concord and inversion, which Wolfram (2004) classified as " stable" urban AAE features.

In all cases, the numeric values of the demographic correlation coefficients (including those not shown such as Asian-American population) are so low as to be statistically inconclusive. However, in all AAE/SAE syntax pairs except for the negations, the correlation coefficients for female users showed a tendency toward positive. This could provide support for the female identity-expression hypothesis proposed by Eckert and McConnell-Ginet (2013) but could also indicate an error with the samples obtained using the current AAE syntax patterns (e.g. smaller samples tend to skew toward areas with more women). Further comparison of male vs. female AAE usage is necessary to provide more evidence for the apparent tendency toward women.

5 Conclusion and Future Directions

This thesis proposes (a) a method for detecting AAE syntactic constructions in tweets, and (b) using the metadata from said tweets to approximate the demographics of the users of AAE constructions. The goal of this thesis is to estimate the current state of AAE usage among American social media users. This project has not yet uncovered a clear connection between ethnic demographics and the use of AAE syntax, suggesting that the dialect is more widespread than previous studies such as Wood and Zanuttini (2014) may have predicted. However, several analyses of the data have suggested that women on Twitter employ AAE syntax more than men, even taking into consideration the slightly higher proportion of women using social media. A different approach to data analysis, and potentially stricter syntax-detection patterns (e.g. only detecting special sub-cases of copula deletion), will be necessary to discover trends of AAE usage within the massive dataset.

Since the synchronic approach seemed to yield limited results, the next step in the project will be analyzing the data on a diachronic scale. The first goal of this approach is to corroborate or challenge the claims of Wolfram (2004) concerning

Table 3: Regression Results over AAE and SAE Demographics

AAE/SAE Syntax Pair	SAE User %	Coefficient (Population)	Coefficient (%Caucasian)	Coefficient (%African-American)	Coefficient (Female)
Copula Deletion/ Standard Copula	93.30%	0.0208	−0.0001	−0.0005	0.0321
Future *finna*/ Future *gonna*	61.75%	0.0312	−0.0024	−0.0006	0.0458
Habitual *be*/ Standard *be*+V_{ing}	79.79%	0.0361	−0.0032	−0.0019	0.0529
Continuative *steady*/ Standard *be*+V_{ing}	79.79%	0.0669	−0.0077	−0.0027	0.0505
Completive *done*/ Standard V_{PST}	94.12%	0.0846	−0.0076	−0.0045	0.0685
Negative Concord/ Standard Negation	22.15%	0.0091	0.0009	0.0014	−0.0006
Negative Inversion/ Non-Inverted Negation	20.16%	−0.0181	0.0005	0.0006	0.0018

“intensifying,” “stable,” and “receding” AAE syntax features by extrapolating a larger pattern of change from the limited time series available (July - December 2013). Secondarily, assuming that some of these features are changing in usage over time, this approach will test whether female Twitter users are leaders of change-in-progress, a trend proven by previous sociolinguistic studies (Eckert and McConnell-Ginet, 2013). In contrast, Reyes (2005) proposes that Asian-American young men adopt AAE slang to emulate African American “hyper-masculinity”, a trend which could lead to men rather than women being leaders of dialect reappropriation. To discover such trends of adoption among individual users, it may also make sense to track each tweeter's AAE vs. SAE usage to determine the extent to which an individual user's syntax can change over time.

Outside the scope of this study, future work might consider using a semi-supervised training method over POS n-grams to automatically detect certain syntactic constructions. This would eliminate the need for rigid regular expressions in searching for tweets with AAE syntax, and also enable the detection of a variety of other constructions. In addition, future AAE studies in Twitter may benefit from the approach of Bergsma et al. (2013), which use user names and patterns of interaction to infer “hidden properties” such as gender and race. Under this framework, researchers might leverage online social media metadata to explore emergent linguistic behavior of various speech communities linked by patterns of interaction. This is an intriguing possibility to consider with the increasing presence of online communities like “Black Twitter” (Sharma, 2013), which allow real-world linguistic trends like AAE syntax to propagate in virtual space.

Acknowledgments

This study was funded in part by the Neukom Institute for Computational Science (`http://neukom.dartmouth.edu/`) and the Presidential Scholars Program at Dartmouth. The project is being supervised by Professor James Stanford and postdoctorate scholar Dr. Sravana Reddy of Dartmouth College.

References

Shane Bergsma, Mark Dredze, Benjamin van Durme, Theresa Wilson, and David Yarowsky. 2013. Broadly Improving User Classification via Communication-Based Name and Location Clustering on Twitter. *Proceedings of NAACL-HLT 2013*, pages 1010–1019.

Leon Derczynski, Alan Ritter, Sam Clark, and Kalina Bontcheva. 2013. Twitter Part-of-Speech Tagging for All: Overcoming Sparse and Noisy Data. *Proceedings of Recent Advances in Natural Language Processing*, pages 198–206.

Maeve Duggan and Joanna Brenner. 2013. The Demographics of Social Media Users - 2012. *Pew Re-*

search Center's Internet & American Life Project, pages 1–14.

Penelope Eckert and Sally McConnell-Ginet. 2013. *Language and Gender*. Cambridge University Press, New York, 2 edition.

Jacob Eisenstein, Brendan O'Connor, Noah Smith, and Eric Xing. 2010. A latent variable model for geographic lexical variation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jacob Eisenstein, Noah A. Smith, and Eric P. Xing. 2011. Discovering Sociolinguistic Associations with Structured Sparsity. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, 1(49):1365–1374.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments. *Proceedings of NAACL-HLT*, pages 380–390.

Lisa Green and Thomas Roeper. 2007. The Acquisition Path for Tense-Aspect: Remote Past and Habitual in Child African American English. *Language Acquisition*, 14(3):269–313.

Lisa Green. 2002. *African American English: A Linguistic Introduction*. Cambridge University Press, Cambridge.

Tyler Kendall, Joan Bresnan, and Gerard van Herk. 2011. The dative alternation in African American English: Researching syntactic variation and change across sociolinguistic datasets. *Corpus Linguistics and Linguistics Theory*, 7(2):229–244.

William Labov, Sharon Ash, and Charles Boberg. 2005. *The Atlas of North American English: Phonetics, Phonology, and Sound Change*. Walter de Gruyer, Berlin.

Wiliam Labov. 2012. *Language in the inner city: Studies in the black English vernacular*. University of Philadelphia Press, Philadelphia, PA.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2013. Improved Part-of-Speech Tagging for Online Conversational Text with Word Clusters. *Proceedings of NAACL-HLT 2013*, pages 380–390.

Angela Reyes. 2005. Appropriation of African American slang by Asian American youth. *Journal of Sociolinguistics*, 9(4):509–532.

Sanjay Sharma. 2013. Black Twitter?: Racial Hashtags, Networks and Contagion. *new formations: a journal of culture/theory/politics*, 78(1):46–64.

Luke Sloan, Jeffrey Morgan, William Housley, Matthew Williams, Adam Edwards, Pete Burnap, and Omer Rana. 2013. Knowing the Tweeters: Deriving Sociologically Relevant Demographics from Twitter. *Sociological Research Online*, 18(3).

Walt Wolfram. 2004. Urban African American Vernacular English: morphology and syntax*. In Bernard Kortmann, editor, *A handbook of varieties of English. 1. Phonology, Volume 2*, volume 2, pages 319–340. Walter de Gruyer.

Jim Wood and Natalie Zanuttini. 2014. The Yale Grammatical Diversity Project. http://microsyntax.sites.yale.edu/.

Malcah Yaeger-Dror and Erik R. Thomas. 2010. *African American English Speakers and Their Participation in Local Sound Changes: A Comparative Study*. Duke University Press for the American Dialect Society, Durham, NC.

Expanding the Range of Automatic Emotion Detection in Microblogging Text

Jasy Liew Suet Yan
School of Information Studies
Syracuse University
Syracuse, New York, USA
jliewsue@syr.edu

Abstract

Detecting emotions on microblogging sites such as Twitter is a subject of interest among researchers in behavioral studies investigating how people react to different events, topics, etc., as well as among users hoping to forge stronger and more meaningful connections with their audience through social media. However, existing automatic emotion detectors are limited to recognize only the basic emotions. I argue that the range of emotions that can be detected in microblogging text is richer than the basic emotions, and restricting automatic emotion detectors to identify only a small set of emotions limits their practicality in real world applications. Many complex emotions are ignored by current automatic emotion detectors because they are not programmed to seek out these "undefined" emotions. The first part of my investigation focuses on discovering the range of emotions people express on Twitter using manual content analysis, and the emotional cues associated with each emotion. I will then use the gold standard data developed from the first part of my investigation to inform the features to be extracted from text for machine learning, and identify the emotions that machine learning models are able to reliably detect from the range of emotions which humans can reliably detect in microblogging text.

1 Introduction

The popularity of microblogging sites such as Twitter provide us with a new source of data to study how people interact and communicate with their social networks or the public. Emotion is a subject of interest among researchers in behavioral studies investigating how people react to different events, topics, etc., as well as among users hoping to forge stronger and more meaningful connections with their audience through social media. There is growing interest among researchers to study how emotions on social media affect stock market trends (Bollen, Mao, & Zeng, 2011), relate to fluctuations in social and economic indicators (Bollen, Pepe, & Mao, 2011), serve as a measure for the population's level of happiness (Dodds & Danforth, 2010), and provide situational awareness for both the authorities and the public in the event of disasters (Vo & Collier, 2013).

In order to perform large-scale analysis of emotion phenomena and social behaviors on social media, there is a need to first identify the emotions that are expressed in text as the interactions on these platforms are dominantly text-based. With the surging amount of emotional content on social media platforms, it is an impossible task to detect the emotions that are expressed in each message using manual effort. Automatic emotion detectors have been developed to deal with this challenge. However, existing applications still rely on simple keyword spotting or lexicon-based methods due to the absence of sufficiently large emotion corpora for training and testing machine learning models

Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 38–44,
Gothenburg, Sweden, April 26-30 2014. ©2014 Association for Computational Linguistics

(Bollen, Pepe, et al., 2011; Dodds & Danforth, 2010).

Research in using machine learning techniques to process emotion-laden text is gaining traction among sentiment analysis researchers, but existing automatic emotion detectors are restricted to identify only a small set of emotions, thus limiting their practicality for capturing the richer range of emotions expressed on social media platforms. The current state-of-the-art of simply adopting the basic emotions described in the psychology literature as emotion categories in text, as favored by a majority of scholars, is too limiting. Ekman's six basic emotions (happiness, sadness, fear, anger, disgust, and surprise) (Ekman, 1971) are common emotion categories imposed on both humans and computers tasked to detect emotions in text (Alm, Roth, & Sproat, 2005; Aman & Szpakowicz, 2007; Liu, Lieberman, & Selker, 2003). It is important to note that most basic emotions such as the six from Ekman are derived from facial expressions that can be universally recognized by humans. Verbal expressions of emotion are different from non-verbal expressions of emotion. Emotions expressed in text are richer than the categories suggested by the basic emotions. Also, people from different cultures use various cues to express a myriad of emotions in text.

By using a restricted set of emotion categories, many emotions not included as part of the basic set are ignored or worse still, force-fitted into one of the available emotion categories. This introduces a greater level of fuzziness in the text examples associated with each emotion.

Example [1]: *"My prayers go to family of Amb. Stevens & others affected by this tragedy. We must not allow the enemy to take another. http://t.co/X8xTzeE4"*

Example [1] is an obvious case of "sympathy" as the writer is expressing his or her condolences to people affected by a tragedy. If "sympathy" is not in the pre-defined list of emotion categories that humans can choose from, human annotators may label this instance as "sadness", which is not entirely accurate. These inaccuracies will then be propagated into the automatic emotion detector.

While the basic emotions have been established as universal emotions (Ekman, 1999), their usefulness in emotion detection in text is still unclear. How useful are the six basic emotions in detecting consumers' emotional reactions towards a product or service from microblogs? What if a company wishes to detect disappointment? The focus on only the basic emotions has resulted in a dearth of effort to build emotion detectors that are able to recognize a wider range of emotions, especially the complex ones. Complex emotions are not merely combinations of the basic ones. For example, none of the combinations of Ekman's six basic emotions seem to represent "regret" or "empathy". Without human-annotated examples of complex emotions, automatic emotion detectors remain ignorant of these emotions simply because they are not programmed to seek out these "undefined" emotions.

There is a need to create automatic emotion detectors that can detect a richer range of emotions apart from the six basic emotions proposed by Ekman to deal with emotional content from social media platforms. A broader range of emotions will enable automatic emotion detectors to capture more fine-grained emotions that truly reflect actual human emotional experience. Limited research has been done so far to determine the full range of emotions which humans can reliably detect in text, as well as salient cues that can be used to identify distinct emotions in text. A crucial step to address this gap is to develop a gold standard corpus annotated with a richer set of emotions for machine learning models to learn from.

My research goal is to first discover the range of emotions humans can reliably detect in microblogging text, and investigate specific cues humans rely on to detect each emotion. Is there a universal set of cues humans rely on to detect a particular emotion or do these cues differ across individuals? Using grounded theory, the first part of my investigation focuses on discovering the range of emotions from tweets collected from a popular microblogging site, Twitter, and the emotional cues associated with each emotion. Twitter offers a wealth of publicly available emotional content generated by a variety of users on numerous topics. The inherently social nature of interactions on Twitter also allows me to investigate social emotions apart from personal emotions. In the second part of my investigation, human annotations from the first part of my investigation will serve as gold standard data for machine learning experiments used to determine the emotions that automatic methods can reliably detect from the range of emotions that humans can reliably identify.

2 Background

Early research on automatic emotion detection in text is linked to subjectivity analysis (Wiebe, Wilson, Bruce, Bell, & Martin, 2004; Wiebe, Wilson, & Cardie, 2005). Emotion detection in text is essentially a form of sentiment classification task based on finer-grained emotion categories. Automatic emotion detection has been applied in the domain of emails (Liu et al., 2003), customer reviews (Rubin, Stanton, & Liddy, 2004), children's stories (Alm et al., 2005), blog posts (Aman & Szpakowicz, 2007), newspaper headlines (Strapparava & Mihalcea, 2008), suicide notes (Pestian et al., 2012), and chat logs (Brooks et al., 2013). Early development of automatic emotion detectors focused only on the detection of Ekman's six basic emotions: happiness, surprise, sadness, fear, disgust, and anger (Alm et al., 2005; Aman & Szpakowicz, 2007; Liu et al., 2003; Strapparava & Mihalcea, 2008). Plutchik's model is an expansion of Ekman's basic emotions through the addition of trust and anticipation in his eight basic emotions (Plutchik, 1962), while Izard's ten basic emotions also include guilt and shame (Izard, 1971).

Scholars have only recently started to expand the categories for automatic emotion classification as noted in the 14 emotions that are pertinent in the domain of suicide notes (Pestian et al., 2012), and 13 top categories that are used for emotion classification out of 40 emotions that emerged from the scientific collaboration chat logs (Brooks et al., 2013; Scott et al., 2012). However, existing gold standard corpora are limited by the emotion categories that are most often specific to a particular domain. Furthermore, it is difficult to pinpoint the exact words, symbols or phrases serving as salient emotion indicators because existing gold standard data are manually annotated at the sentence or message level.

Using Twitter, scholars have explored different strategies to automatically harness large volumes of data automatically for emotion classification. Pak & Paroubek (2010) applied a method similar to Read (2005) to extract tweets containing happy emoticons to represent positive sentiment, and sad emoticons to represent negative sentiment. First, this limits the emotion classifier to detect only happiness and sadness. Second, the lack of clear distinctions between the concepts of sentiment and emotion is problematic because tweeters may express a negative emotion towards an entity which they hold a positive sentiment on, and vice versa. For example, a tweeter expressing sympathy to another person who has experienced an unfortunate event is expressing a negative emotion but the tweet contains an overall positive sentiment. Third, such a data collection method assumes that the emotion expressed in the text is the same as the emotion the emoticon represents, and does not take into account of cases where the emotion expressed in the text may not be in-sync with the emotion represented by the emoticon (e.g., sarcastic remarks).

Mohammad (2012) and Wang, Chen, Thirunarayan, & Sheth (2012) applied a slightly improved method to create a large corpus of readily-annotated tweets for emotion classification. Twitter allows the use of hashtags (words that begin with the # sign) as topic indicators. These scholars experimented with extracting tweets that contain a predefined list of emotion words appearing in the form of hashtags. Mohammad (2012) only extracted tweets with emotion hashtags corresponding to Ekman's six basic emotions (#anger, #disgust, #fear, #joy, #sadness, and #surprise) while Wang et al. (2012) expanded the predefined hashtag list to include emotion words associated with an emotion category, as well as the lexical variants of these emotion words. Although this method allows researchers to take advantage of the huge amount of data available on Twitter to train machine learning models, little is known about the specific emotional cues that are associated with these emotion categories. Also, this data collection method is biased towards tweeters who choose to express their emotions explicitly in tweets.

Kim, Bak, & Oh (2012) proposed a semi-supervised method using unannotated data for emotion classification. They first applied Latent Dirichlet Allocation (LDA) to discover topics from tweets, and then determined emotions from the discovered topics by calculating the pointwise mutual information (PMI) score for each emotion from a list of eight emotions given a topic. The evaluation of this method using a corpus of manually annotated tweets revealed that this automatic emotion detector only managed to correctly classify 30% of tweets from the test dataset. The gold standard corpus used for evaluation was developed through manual annotations using Amazon Mechanical Turk (AMT). Only 3% of the tweets received full agreement among five annotators.

3 Defining Emotions In Text

In everyday language, people refer to emotion as prototypes of common emotions such as happiness, sadness, and anger (Fehr & Russell, 1984). In the scientific realm, emotion is generally defined as "ongoing states of mind that are marked by mental, bodily or behavioral symptoms" (Parrott, 2001). Specifically, each emotion category (e.g., happiness, sadness, anger, etc.) is distinguishable by a set of mental, bodily or behavioral symptoms. When a person expresses emotion in text, these symptoms are encoded in written language (words, phrases and sentences).

Emotion in text is conceptualized as emotion expressed by the writer of the text. Emotion expression consists of "signs that people give in various emotional states", usually with the intention to be potentially perceived or understood by the others (Cowie, 2009). People express their emotional states through different non-verbal (e.g., facial expression, vocal intonation, and gestures) and verbal (e.g., text, spoken words) manifestations. Emotion expression in text is a writer's descriptions of his or her emotional experiences or feelings. It is important to note that emotion expression only provides a window into a person's emotional state depending on what he or she chooses to reveal to the others. It may not be depictions of a person's actual emotional state, which is a limitation to the study of emotion in text (Calvo & D'Mello, 2010).

4 Research Questions

Detecting emotions in microblog posts poses new challenges to existing automatic emotion detectors due to reasons described below:

- Unlike traditional texts, tweets consist of short texts expressed within the limit of 140 characters, thus the language used to express emotions differs from longer texts (e.g., blogs, news, and fairy tales).

- The language tweeters use is typically informal. Automatic emotion detectors must be able to deal with the presence of abbreviations, acronyms, orthographic elements, and misspellings.

- Emotional cues are not limited to only emotion words. Twitter features such as #hashtags (topics), @username, retweets, and other user profile metadata may serve as emotional cues.

Using data from Twitter, a popular microblogging platform, I will develop an initial framework to study the richness of emotions expressed for personal, as well as for social purposes. My research investigation is guided by the research questions listed below:

- What emotions can humans reliably detect in microblogging text?
- What salient cues are associated with each emotion?
- How can good features for machine learning be identified from the salient cues humans associate with each emotion?
- What emotions in microblogging text can be reliably detected using current machine learning techniques?

5 Proposed Methodology

My research design consists of three phases: 1) small-scale inductive content analysis for code book development, 2) large-scale deductive content analysis for gold standard data development, and 3) the design of machine learning experiments for automatic emotion detection in text.

5.1 Data Collection

When sampling for tweets from Twitter, I will utilize three sampling strategies to ensure the variability of emotions being studied. First, I will collect a random sample of publicly-available tweets. This sampling strategy aims to create a sample that is representative of the population on Twitter but may not produce a collection of tweets with sufficient emotional content. The second sampling strategy is based on topics or events. To ensure that tweets are relevant to this investigation, tweets will be sampled based on hashtags of events likely to evoke text with emotional content. Topics will include politics, sports, products/services, festive celebrations, and disasters.

The third sampling strategy is based on users. This sampling strategy allows me to explore the range of emotions expressed by different individuals based on different stimuli, and not biased towards any specific events. To make the manual annotation feasible, I plan to first identify the usernames of 1) active tweeters with a large number of followers (e.g., tweets from politicians) to ensure sufficient data for analysis, and 2) random tweeters to represent “average” users of Twitter. I acknowledge that this sampling strategy may be limited to only certain groups of people, and may not be representative of all Twitter users but it offers a good start to exploring the range of emotions being expressed in individual streams of tweets.

5.2 Phase 1

To develop a coding scheme for emotion annotation, I will first randomly sample 1,000 tweets each from the random, topic-based, and user-based datasets for open coding. I will work with a small group of coders to identify the emotion categories from a subset of the 1,000 tweets. Coders will be given instructions to assign each tweet with only one emotion label (i.e., the best emotion tag to describe the overall emotion expressed by the writer in a tweet), highlight the specific cues associated with the emotion, as well as identify the valence and intensity of the emotion expressed in the tweet.

To verify the grouping of the emotion tags, coders will be asked to perform a card sorting exercise to group emotion tags that are semantically similar in the same group. Based on the discovered emotion categories, nuanced colorations within each category may be detected from the valence and intensity codes.

Coders will incrementally annotate more tweets (300 tweets per round) until a point of saturation is reached, where new emotion categories stop emerging from data. I will continuously meet with the coders to discuss disagreements until the expected inter-annotator agreement threshold for the final set of emotion categories is achieved.

5.3 Phase 2

Using the coding scheme developed from Phase 1, I will obtain a larger set of manual annotations using Amazon Mechanical Turk (AMT). AMT allows me to collect manual annotations of emotions on a large-scale, thus enabling me to investigate if there are any differences as to what a larger crowd of people identify as emotion cues in tweets. Each tweet will be annotated by at least three coders. To ensure the quality of the manual annotations collected from AMT, workers on AMT will have to undergo a short training module explaining the coding scheme, and will have to pass a verification test before being presented with the actual tweets to be annotated. Inter-annotator agreement will be calculated, and the emotion categories that humans can reliably detect in text will be identified.

5.4 Phase 3

Detecting a single emotion label for each tweet can be defined as a multi-class classification problem. The corpus from Phase 2 will be used as training data, and the corpus from Phase 1 will be used as testing data for the machine learning model. An analysis of the emotional cues from Phase 1 and Phase 2 datasets is conducted to identify salient features to be used for machine learning. Support vector machines (SVM) have been shown to perform well in this problem space (Alm et al., 2005; Aman & Szpakowicz, 2007; Brooks et al., 2013; Cherry, Mohammad, & de Bruijn, 2012) so I will run experiments using SVM, and compare the performance of the model against a baseline using simple lexical features (i.e., n-grams).

6 Research Contributions

Analyzing the emotional contents in tweets can expand the theoretical understanding of the range of emotions humans express on social media platforms like Twitter. From a natural language processing standpoint, it is also crucial for the community to gain clearer insights on the cues associated with each fine-grained emotion. On top of that, findings from the machine learning experiments will inform the community as to whether training the machine learning models based on data collected using usernames, instead of topic hashtags will reduce noise in the data, and improve the performance of automatic emotion detection in microblogging texts.

The expected contributions of this research investigation are three-fold: 1) the construction of an emotion taxonomy and detailed annotation scheme that could provide a useful starting point for future research, 2) the creation of machine learning models that can detect a wider range of emotions in text in order to enable researchers to tap into this wealth of information provided by Twitter to study a greater multitude of behavioral and social phenomenon, and 3) findings on the range of emotions people express on Twitter can potentially help inform the design of social network platforms to be more emotion sensitive.

References

Alm, C. O., Roth, D., & Sproat, R. (2005). Emotions from text: Machine learning for text-based emotion prediction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing* (pp. 579–586). Stroudsburg, PA, USA.

Aman, S., & Szpakowicz, S. (2007). Identifying expressions of emotion in text. In *Text, Speech and Dialogue* (pp. 196–205).

Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, *2*(1), 1–8.

Bollen, J., Pepe, A., & Mao, H. (2011). Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media* (pp. 450–453).

Brooks, M., Kuksenok, K., Torkildson, M. K., Perry, D., Robinson, J. J., Scott, T. J., … Aragon, C. R. (2013). Statistical affect detection in collaborative chat. Presented at the Conference on Computer Supported Cooperative Work and Social Computing, San Antonio, TX.

Calvo, R. A., & D’Mello, S. (2010). Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on Affective Computing*, *1*(1), 18–37.

Cherry, C., Mohammad, S. M., & de Bruijn, B. (2012). Binary classifiers and latent sequence models for emotion detection in suicide notes. *Biomedical Informatics Insights*, *5*, 147–154.

Cowie, R. (2009). Perceiving emotion: Towards a realistic understanding of the task. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, *364*(1535), 3515–3525.

Dodds, P. S., & Danforth, C. M. (2010). Measuring the happiness of large-scale written expression: Songs, blogs, and Presidents. *Journal of Happiness Studies*, *11*(4), 441–456.

Ekman, P. (1971). Universals and cultural differences in facial expressions of emotion. *Nebraska Symposium on Motivation*, *19*, 207–283.

Ekman, P. (1999). Basic emotions. In *Handbook of Cognition and Emotion* (pp. 45–60). John Wiley & Sons, Ltd.

Fehr, B., & Russell, J. A. (1984). Concept of emotion viewed from a prototype perspective. *Journal of Experimental Psychology: General*, *113*(3), 464–486.

Izard, C. E. (1971). *The face of emotion* (Vol. xii). East Norwalk, CT, US: Appleton-Century-Crofts.

Kim, S., Bak, J., & Oh, A. H. (2012). Do you feel what I feel? Social aspects of emotions in Twitter conversations. In *International AAAI Conference on Weblogs and Social Media (ICWSM)*.

Liu, H., Lieberman, H., & Selker, T. (2003). A model of textual affect sensing using real-world knowledge. In *Proceedings of the 8th International Conference on Intelligent User Interfaces* (pp. 125–132).

Mohammad, S. M. (2012). #Emotional tweets. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics*. Montreal, QC.

Pak, A., & Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *Seventh International Conference on Language Resources and Evaluation (LREC)*.

Parrott, W. G. (2001). *Emotions in social psychology: Essential readings* (Vol. xiv). New York, NY, US: Psychology Press.

Pestian, J. P., Matykiewicz, P., Linn-Gust, M., South, B., Uzuner, O., Wiebe, J., … Brew, C. (2012). Sentiment analysis of suicide notes: A shared task. *Biomedical Informatics Insights*, *5*(Suppl. 1), 3–16.

Plutchik, R. (1962). *The Emotions: Facts, theories, and a new model*. New York: Random House.

Read, J. (2005). Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL Student Research Workshop* (pp. 43–48). Stroudsburg, PA, USA.

Rubin, V. L., Stanton, J. M., & Liddy, E. D. (2004). Discerning emotions in texts. In *The AAAI Symposium on Exploring Attitude and Affect in Text (AAAI-EAAT)*.

Scott, T. J., Kuksenok, K., Perry, D., Brooks, M., Anicello, O., & Aragon, C. (2012). Adapting grounded theory to construct a taxonomy of affect in collaborative online chat. In *Proceedings of the 30th ACM International Conference on Design of Communication* (pp. 197–204). New York, USA.

Strapparava, C., & Mihalcea, R. (2008). Learning to identify emotions in text. In *Proceedings of the 2008 ACM Symposium on Applied Computing* (pp. 1556–1560). New York, USA.

Vo, B.-K. H., & Collier, N. (2013). Twitter emotion analysis in earthquake situations. *International Journal of Computational Linguistics and Applications*, *4*(1), 159–173.

Wang, W., Chen, L., Thirunarayan, K., & Sheth, A. P. (2012). Harnessing Twitter “big data” for automatic emotion identification. In *2012 International Conference on Privacy, Security, Risk and Trust (PASSAT), and 2012 International Conference on Social Computing (SocialCom)* (pp. 587–592).

Wiebe, J. M., Wilson, T., Bruce, R., Bell, M., & Martin, M. (2004). Learning subjective language. *Computational Linguistics*, *30*(3), 277–308.

Wiebe, J. M., Wilson, T., & Cardie, C. (2005). Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, *39*(2-3), 165–210.

Resolving Coreferent and Associative Noun Phrases in Scientific Text

Ina Rösiger
Institute for Natural Language Processing
University of Stuttgart, Germany
Pfaffenwaldring 5b, 70569 Stuttgart
roesigia@ims.uni-stuttgart.de

Simone Teufel
Computer Laboratory
University of Cambridge, UK
15 JJ Thomson Avenue, Cambridge CB3 0FD
sht25@cl.cam.ac.uk

Abstract

We present a study of information status in scientific text as well as ongoing work on the resolution of coreferent and associative anaphora in two different scientific disciplines, namely computational linguistics and genetics. We present an annotated corpus of over 8000 definite descriptions in scientific articles. To adapt a state-of-the-art coreference resolver to the new domain, we develop features aimed at modelling technical terminology and integrate these into the coreference resolver. Our results indicate that this integration, combined with domain-dependent training data, can outperform the performance of an out-of-the-box coreference resolver. For the (much harder) task of resolving associative anaphora, our preliminary results show the need for and the effect of semantic features.

1 Introduction

Resolving anaphoric relations automatically requires annotated data for training and testing. Anaphora and coreference resolution systems have been tested and evaluated on different genres, mainly news articles and dialogue. However, for scientific text, annotated data are scarce and coreference resolution systems are lacking (Schäfer et al., 2012). We present a study of anaphora in scientific literature and show the difficulties that arise when resolving coreferent and associative entities in two different scientific disciplines, namely computational linguistics and genetics.

Coreference resolution in scientific articles is considered difficult due to the high proportion of definite descriptions (Watson et al., 2003), which typically require domain knowledge to be resolved. The more complex nature of the texts is also reflected in the heavy use of abstract entities such as results or variables, while easy-to-resolve named entities are less frequently used. We test an existing, state-of-the-art coreference resolution tool on scientific text, a domain on which it has not been trained, and adapt it to this new domain. We also address the resolution of *associative anaphora* (Clark, 1975; Prince, 1981), a related phenomenon, which is also called bridging anaphora. The interpretation of an associative anaphor is based on the associated antecedent, but the two are not coreferent. Examples 1 and 2 show two science-specific cases of associative anaphora from our data.

(1) Xe-Ar was found to be in *a layered structure* with Ar on **the surface**[1].

(2) We base our experiments on *the Penn treebank*. **The corpus size** is ...

The resolution of associative links is important because it can help in tasks which use the concept of textual coherence, e.g. Barzilay and Lapata (2008)'s entity grid or Hearst (1994)'s text segmentation. They might also be of use in higher-level text understanding tasks such as textual entailment (Mirkin et al., 2010) or summarisation based on argument overlap (Kintsch and van Dijk, 1978; Fang and Teufel, 2014).

Gasperin (2009) showed that biological texts differ considerably from other text genres, such as news text or dialogue. In this respect, our results confirm that the proportion between non-referring and referring entities in scientific text differs from that reported for other genres. The same holds for the type and relative number of linguistic expressions used for reference. To address this issue, we decided to investigate *information status* (Nissim et al., 2004) of noun phrases. Information status tells us whether a noun phrase refers to an already

[1] Anaphors are typed in bold face, their antecedents shown in italics.

Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 45–55,
Gothenburg, Sweden, April 26-30 2014.

known entity, or whether it can be treated as non-referring. Since no corpus of full-text scientific articles annotated with both information status and anaphoric relations was available, we had to create and annotate our own corpus. The main contributions of this work are (i) a new information status-based annotation scheme and an annotated corpus of scientific articles, (ii) a study of information status in scientific text that compares the distribution of the different categories in scientific text with the distribution in news text, as well as between the two scientific disciplines, (iii) experiments on the resolution of coreferent anaphora: we devise domain adaptation for science and show how this improves an out-of-the-box coreference resolver, and (iv) experiments on the resolution of associative anaphora with a coreference resolver that is adapted to this new notion of "reference" by including semantic features. To the best of our knowledge, this is the first work on anaphora resolution in multi-discipline, full-text scientific papers that also deals with associative anaphora.

2 Related Work

Noun phrase coreference resolution is the task of determining which noun phrases (NPs) in a text or dialogue refer to the same real-world entities (Ng, 2010). Resolving anaphora in scientific text has only recently gained interest in the research community and focuses mostly on the biomedical domain (Gasperin, 2009; Batista-Navarro and Ananiadou, 2011; Cohen et al., 2010). Some work has been done for other disciplines, such as computational linguistics. Schäfer et al. (2012) present a large corpus of 266 full-text computational linguistics papers from the ACL Anthology, annotated with coreference links. The CoNLL shared task 2012 on modelling multilingual unrestricted coreference in OntoNotes (Pradhan et al., 2012) produced several state-of-the-art coreference systems (Fernandes et al., 2012; Björkelund and Farkas, 2012; Chen and Ng, 2012) trained on news text and dialogue, as provided in the OntoNotes corpus (Hovy et al., 2006). Other state-of-the-art systems, such as Raghunathan et al. (2010) and Berkeley's Coreference Resolution System (Durrett and Klein, 2013), also treat coreference as a task on news text and dialogue. We base our experiments on the IMS coreference resolver by Björkelund and Farkas (2012), one of the best publicly available English coreference systems. The resolver uses the decision of a cluster-based decoding algorithm, i.e. one that decides whether two mentions are placed in the same or in different clusters, or whether they should be considered singletons. Their novel idea is that the decision of this algorithm is encoded as a feature and fed to a pairwise classifier, which makes decisions about pairs of mentions rather than clusters. This stacked approach overcomes problems of previous systems that are based on the isolated pairwise decision. The features used are mostly taken from previous work on coreference resolution and encode a variety of information, i.e, surface forms and their POS tags, subcategorisation frames and paths in the syntax tree as well as the semantic distance between the surface forms (e.g. edit distance).

However, none of this work is concerned with associative anaphora. Hou et al. (2013) present a corpus of news text annotated with associative links that are not limited with respect to semantic relations between anaphor and antecedent. Their experiments focus on antecedent selection only, assuming that the recognition of associative entities has already been performed. Information status has been investigated extensively in different genres such as news text, e.g. in Markert et al. (2012). Poesio and Vieira (1998) performed an information status-based corpus study on news text, defining the following categories: coreferential, bridging, larger situation, unfamiliar and doubt. To the best of our knowledge, there is currently no study on information status in scientific text.

In this paper, we propose a classification scheme for scientific text that is derived from Riester et al. (2010) and Poesio and Vieira (1998). We investigate the differences between news text and scientific text by analysing the distribution of information status categories. We hypothesise that the proportion of associative anaphora in scientific text is higher than in news text, making it necessary to resolve them in some form. Our experiments on the resolution of coreferent anaphora concern the domain-adaptation of a coreference resolver to this new domain and examine the effect of domain-dependent training data and features aimed at capturing technical terminology. We also present an unusual setup where we assume that an existing coreference resolver can also be used to identify associative links. We integrate semantic features in the hope of detecting cases where domain knowledge is required to establish the relation between the anaphor and the antecedent.

	Category	Example
COREFERENCE LINKS	GIVEN (SPECIFIC)	We present *the following experiment.* $\boxed{\textbf{It}}$ deals with ...
	GIVEN (GENERIC)	We use *the Jaccard similarity coefficient* in our experiments. $\boxed{\textbf{The Jaccard similarity coefficient}}$ is useful for ...
ASSOCIATIVE LINKS	ASSOCIATIVE	Xe-Ar was found to be in *a layered structure* with Ar on $\boxed{\textbf{the surface}}$.
Categories without links	ASSOCIATIVE (SELF-CONTAINING)	**The structure of the protein** ...
	DESCRIPTION	**The fact that the accuracy improves** ...
	UNUSED	**Noam Chomsky** introduced the notion of ...
	DEICTIC	**This experiment** deals with ...
	PREDICATIVE	Pepsin, **the enzyme, ...**
	IDIOM	On **the one hand** ... on **the other hand** ...
	DOUBT	

Table 1: Categories in our classification scheme

3 Corpus Creation

We manually annotated a small scientific corpus to provide a training and test corpus for our experiments, using the annotation tool Slate (Kaplan et al., 2012).

3.1 Annotation Scheme

Two types of reference are annotated, namely COREFERENCE and ASSOCIATIVE LINKS. COREFERENCE LINKS are annotated for all types of nominal phrases; such links are annotated between enitites that refer to the same referent in the real world. ASSOCIATIVE LINKS and information status categories are only annotated for definite noun phrases. In our scheme, ASSOCIATIVE LINKS are only annotated when there is a clear relation between the two entities. As we do not pre-define possible associative relations, this definition is vague, but it is necessary to keep the task as general as possible. Additionally, we distinguish the following nine categories, as shown in Table 1[2]: The category GIVEN comprises coreferent entities that refer back to an already introduced entity. If a coreference link is detected, the referring expression is marked as GIVEN and the link with its referent NP is annotated. The obligatory attribute GENERIC tells us whether the given entity has a generic or a specific reading. ASSOCIATIVE refers to entities that are not coreferent but whose interpretation is based on a previously introduced entity. A typical relation between the two noun phrases is meronymy, but as mentioned above we do not pre-define a set of allowed semantic relations. The category ASSOCIATIVE (SELF-CONTAINING) comprises cases where we identify an associative relation between the head noun phrase and the modifier. ASSOCIATIVE SELF-CONTAINING entities are annotated without a link between the two parts. In scientific text, an entity is considered DEICTIC if it points to an object that is connected to the current text. Therefore, we include all entities that refer to the current paper (or aspects thereof) in this category. Entities that have not been mentioned before and are not related to any other entity in the text, but can be interpreted because they are part of the common knowledge of the writer and the reader are covered by the category UNUSED. DESCRIPTION is annotated for entities that are self-explanatory and typically occur in particular syntactic patterns such as NP complements or relative clauses. Idiomatic expressions or metaphoric use are covered in the category IDIOM. Predicative expressions, including appositions, are annotated as PREDICATIVE. Finally, the category DOUBT is used when the text or the antecedent is unclear. Note that NEW, a category that has been part of most previous classification schemes of information status, is not present as this information status is typically observed in indefinite noun phrases. As we deal exclusively with definite noun phrases[3], we do not include this category in our scheme. In contrast to Poesio and Vieira's scheme, ours contains the additional categories PREDICATIVE, ASSOCIATIVE SELF-CONTAINING, DEICTIC and IDIOM.

[2] The entity being classified is typed in bold face, referring expressions are marked by a box and the referent is shown in italics.

[3] With the exception of coreferring anaphoric expressions, as previously discussed.

	GEN	CL
Sentences	1834	1637
Words	43691	38794
Def. descriptions	3800	4247

Table 2: Properties of the annotated two subcorpora, genetics (GEN) and computational linguistics (CL)

	GEN	CL
Coreference links	1976	2043
Associative links	328	324
Given	1977	2064
Associative	315	280
Associative (sc)	290	272
Description	810	1215
Unused	286	286
Deictic	28	54
Predicative	9	19
Idiom	9	34
Doubt	39	22

Table 3: Distribution of information status categories and links in the two disciplines, in absolute numbers

3.2 Resulting Corpus

Our annotated corpus contains 16 full-text scientific papers, 8 papers for each of the two disciplines. The computational linguistics (CL) papers cover various topics ranging from dialogue systems to machine translation; the genetics (GEN) papers deal mostly with the topic of short interfering RNAs, but focus on different aspects of it. In total, the annotated computational linguistics papers contain 1637 sentences, 38,794 words and 4247 annotated definite descriptions while the annotated genetics papers contain 1834 sentences, 43,691 words and 3800 definite descriptions; the two domain subcorpora are thus fairly comparable in size. See Table 2 for corpus statistics and Table 3 for the distribution of categories and links.

It is well-known that there are large differences in reference phenomena between scientific text and other domains (Gasperin, 2009). In scientific text, it is assumed that the reader has a relatively high level of background. We would expect this general property of scientific text to have an impact on the distribution of categories with respect to information status.

Table 4 compares the two scientific disciplines in our study with each other. We note that the proportion of entities classified as DESCRIPTION in the CL papers is considerably higher than in the GEN papers. The proportions of the other categories are similar, though the proportion of GIVEN, ASSOCIATIVE and UNUSED entities is slightly higher in the GEN articles.

Table 4 also compares the distribution of categories in news text (Poesio and Vieira, 1998; P&V) with that of ours (as far as they are alignable, using our names for categories). Note that on a conceptual level, these categories are equivalent, but there are some differences with respect to the annotation guidelines.

The most apparent difference is the proportion of UNUSED entities (6-7 % in science, 23 % in news text) which might be due to the prevalence of named entities in news text. Compared to the distribution of categories in news text, the proportion of GIVEN entities is about 4-8 % higher in scientific text. The proportion of ASSOCIATIVE entities[4] is twice as high in the scientific domain compared to news text. UNUSED entities have a distinctly lower proportion, with about 7%. As our guidelines limit deictic references to only those that refer to (parts of) the current paper, we get a slightly lower proportion than the 2 % in news text, reported by Poesio and Vieira (1998) in an earlier experiment, where no such limitation was present.

Category	GEN	CL	P&V
Given	52.03	48.60	44.00
Associative	8.29	6.59	8.50
Associative (sc)	7.63	6.40	–
Description	21.31	28.61	21.30
Unused	7.53	6.73	23.50
Deictic	0.74	1.27	–
Predicative	0.24	0.45	–
Idiom	0.24	0.80	(2.00)
Doubt	1.03	0.52	2.60

Table 4: Distribution of information status categories in different domains, in percent

It has been shown in similar annotation experiments on information status, with similarly fine-grained schemes (Markert et al., 2012; Riester et al., 2010), that it is possible to achieve annotation with marginally to highly reliable inter-annotator agreement. In our experiments, only one person (the first author) performed the annotation, so that we cannot compute any agreement measurements. We are currently performing an inter-annotator study with two additional annotators so that we can better judge human agreement and use the annotations as a reliable gold standard.

[4] The union of categories ASSOCIATIVE and ASSOCIATIVE SELF-CONTAINING.

4 Adapting a Coreference Resolver to the Scientific Domain

To show the difficulties that a coreference resolver faces in the scientific domain, we ran, out-of-the-box, a coreference system (Björkelund and Farkas, 2012), that has not been trained on scientific text, on our corpus and perform an error analysis. In particular, we are curious about which of the system's errors are domain-dependent. This analysis motivates a set of terminological features that are incorporated and tested in Section 6.

4.1 Error Analysis

Domain-dependent errors. The lack of semantic, domain-dependent knowledge results in the system's failure to identify coreferent expressions, e.g. those expressed as synonyms. This type of error can be prevented by implementing domain-dependent knowledge. In Example 3, we would like to generate a link between *treebank* and *corpus* as these terms are used as synonyms. The same is true for *protein-forming molecules* and *amino acids* in Example 4.

(3) Experiments were performed with the clean part of *the treebank*. **The corpus** consists of 1 million words.

(4) *Amino acids* are organic compounds made from amine (-NH2) and carboxylic acid (-COOH) functional groups. **The protein-forming molecules** ...

Another common error is that the coreference resolver links all occurrences of demonstrative science-specific expressions such as *this paper* or *this approach* to each other, even if they are several paragraphs apart. In most cases, these demonstrative expressions do not corefer, but refer to an approach or a paper recently described or cited. This type of error is particularly frequent in the computational linguistics domain and might be reduced by a feature that captures this peculiarity. A special case occurs when authors re-use clauses of the abstract in the introduction. The coreference resolver then interprets rather large spans as coreferent which are not annotated in the gold standard. Yet a different kind of error is based on the fact that the coreference resolver has been trained on OntoNotes, i.e. mostly on non-written text. Thus, the classifier has not seen certain phenomena and, for example, links all occurrences of *e.g.* into one equivalence class as it is interpreted as a named entity.

General errors. Some errors are general errors of coreference resolvers in the sense that they have very little to do with domain dependence, such as choosing the wrong antecedent or linking non-referential occurrences of *it* (see Examples 5 and 6).

(5) This approach allows *the processes of building referring expressions* and identifying **their** referents.

(6) *The issue of how to design sirnas that produce high efficacy* is the focus of a lot of current research. Since **it** was discovered that ...

4.2 Terminological Features

This section deals with the design of possible terminological features for our experiments that are aimed at capturing some form of domain knowledge. We create these using the information in 1000 computational linguistics and 1000 genetics papers that are not part of our scientific corpus.

Non-coreferring bias list. Our first feature concentrates on nouns which have a low probability to be coreferring (i.e. category GIVEN) if they appear as the head of noun phrase. We assume that the normal case of coreference between definite noun phrases is that of a concept introduced as an indefinite NP and later referred to as a definite NP, and compile a list of lexemes that do not follow this pattern. NPs with those lexemes should be more likely to be of category UNUSED or DESCRIPTION. We find the lexemes by recording head nouns of definite NPs which are not observed in a prior indefinite NP in the same document (local list) or the entire document collection (global list). We create two lists of such head words for every discipline. The lexemes are arranged in decreasing order of their frequency so that we can use both their presence or non-presence on the list and their rank on the list as potential features.

As can be seen in Table 5, *the presence, the beginning* and *the literature* are definite descriptions that are always used without having been introduced to the discourse. These terms are either part of domain knowledge (*the hearer, the reader*) or part of the general scientific terminology (*the literature*). In the local list we see expressions that can be used without having been introduced, but may in some contexts occur in the indefinite form as well, e.g. *the word* or *the sentence*.

CL		GEN	
(a) global	(b) local	(a) global	(b) local
presence	number	manuscript	data
beginning	word	respect	region
literature	sentence	prediction	gene
hearer	training	monograph	case
reader	user	notion	species

Table 5: Top five terms of local and global non-coreferring bias lists

Collocation list. One of our hypotheses is that the NPs occurring in verb-object collocations are typically not part of any coreference chain. To test this, we use our collection of 2000 scientific papers to extract domain-specific verb-object collocations. We assume that for some collocations, this tendency is stronger (*make use, take place*) than for others that could potentially be coreferring (*see figure, apply rule*). The collocations have been identified with a term extraction tool (Gojun et al., 2012). Every collocation that occurs at least twice in the data is present on the list. Table 6 shows the most frequent terms.

make + use	take + place
give + rise	silence + activity
derive + form	refashion + plan
parse + sentence	predict + sirna
sort + feature	match + predicate
see + figure	use + information
silence + efficiency	follow + transfection
embed + sentence	apply + rule
focus + algorithm	stack + symbol

Table 6: Most frequent occurring collocation candidates in scientific text (unsorted)

Argumentation nouns, work nouns and idioms. As mentioned in Section 4.1, the baseline classifier often links demonstrative, science-specific entities, even if they are several paragraphs apart. To prevent this, we combine a distance measure with a set of 182 argumentation and work nouns taken from Teufel (2010), such as *achievement*, *claim* or *experiment*. We also create a small list of idioms as they are never part of a coreference chain.

5 Adapting a Coreference Resolver for Associative Links in Science

We now turn to the much harder task of resolving associative anaphora.

5.1 Types of Associative Anaphora

To illustrate the different types of associative anaphora, we here show a few examples, mostly taken from the genetics papers. The anaphors are shown in bold face, the antecedents in italics. Many associative anaphors include noun phrases with the same head. In most of these cases, the anaphor contains a different modifier than the antecedent, such as

(8) the negative strain ... **the positive strain**;
(9) three categories ... **the first category**;
(10) siRNAs ... **the most effective siRNAs**.

We assume that these associative relations can be identified with a coreference resolver without adding additional features. Other cases are much harder to identify automatically, such as those where semantic knowledge is required to interpret the relation between the entities:

(11) the classifier ... **the training data**;
(12) this database ... **the large dataset**.

In other cases, the nominal phrase in the antecedent tends to be derivationally related to the head word in the anaphor, as in

(13) the spotty distribution ...**the spots**;
(14) competitor ... **the competitive effect**.

There are also a number of special cases, such as

(15) the one interference process ... **the other interference process**.

We hypothesise that the integration of semantic features discussed in the previous section enables the resolver to cover more than just those cases that are based on the similarity of word forms.

5.2 Semantic Features

It is apparent that the recognition and correct resolution of associative anaphora requires semantic knowledge. Therefore, we adapt the coreference resolver by extending the WordNet feature, one of the features implemented in the IMS resolver, to capture more than just synonyms.

We use the following WordNet relations: Hypernymy (*macromolecule* → *protein*), hyponymy (*nucleoprotein* → *protein*), meronymy (*surface* → *structure*), substance meronymy (*amino acid* → *protein*), topic member (*acute, chronic* → *medicine*) and topic (*periodic table* → *chemistry*).

WordNet's coverage in the scientific domain is surprisingly good: 75,91 % of all common nouns in the GEN papers and 88,12 % in the CL papers are listed in WordNet. Terms that are not covered are, for example, abbreviations of different types of ribonucleic acid in genetics or specialist terms like *tagging, subdialogue* or *SVM* in computational linguistics.

6 Experiments

We now compare the performance of an out-of-the-box coreference system with the resolver trained on our annotated scientific corpus (Section 6.2). We also show the effect of adding additional features aimed at capturing technical terminology. In the experiments on the resolution of associative anaphora (Section 6.3), we test the hypothesis that the coreference resolver is able to adjust to the new notion of reference and show the effect of semantic features.

6.1 Experimental Setup

We perform our experiments using the IMS coreference resolver as a state-of-the-art coreference resolution system (Björkelund and Farkas, 2012)[5]. The algorithm and the features included have not been changed except where otherwise stated. We use the OntoNotes datasets from the CoNLL 2011 shared task[6] (Pradhan et al., 2012; Hovy et al., 2006), only for training the out-of-the-box system. We also use WordNet version 3.0 as provided in the 2012 shared task[7] as well as JAWS, the Java API for WordNet searching[8]. Performance is reported on our annotated corpus, using 8-fold cross-validation and the official CoNLL scorer (version 5).

6.2 Resolving Coreferent References

IMS coreference resolver unchanged. To be able to judge the performance of an existing coreference resolver on scientific text, we first report performance without making any changes to the resolver whatsoever, using different training data. The BASELINE version is trained on the OntoNotes dataset from the CoNLL 2011 shared task. In the SCIENTIFIC version, we only use our annotated scientific papers. MIXED contains the entire OntoNotes dataset as well as the scientific papers, leading to a larger training corpus which compensates for the rather small size of the scientific corpus[9]. Table 7 shows the average CoNLL scores[10] of the two subdomains genetics and computational linguistics.

Training Set	GEN	CL	GEN+CL
Baseline	35.30	40.30	37.80
Scientific	44.94	42.41	43.68
Mixed	47.92	47.44	47.68

Table 7: Resolving coreferent references: CoNLL metric scores for different training sets

The BASELINE achieves relatively low results in comparison to the score of 61.24 that was reported in the shared task (Björkelund and Farkas, 2012). Even though our scientific corpus is only 7% the size of the OntoNotes dataset, it inceases performance of the BASELINE system by 15,6%. The SCIENTIFIC version outperforms the BASELINE version for all of the GEN papers and for 6 out of 8 CL papers. MIXED, the version that combines the scientific corpus with the entire OntoNotes dataset, proves to work best (47.92 for GEN and 47.44 for CL). In THE BASELINE version, the performance on the CL papers is better than on the GEN papers. Interestingly, this is not true for the SCIENTIFIC version, where the performance on the GEN papers is better. However, as the main result here, we can see that training on scientific text was successful. The increase in score in both the SCIENTIFIC and the MIXED version over BASELINE is statistically significant[11]

[5] See: www.ims.uni-stuttgart.de/forschung/ressourcen/werkzeuge/IMSCoref.html
We follow their strategy to use the AMP decoder as the first decoder and the PCF decoder, a pairwise decoder, as a second. The probability threshold is set to 0.5 for the first and 0.65 for the second decoder.

[6] http://conll.cemantix.org/2011/data.html

[7] http://conll.cemantix.org/2012/data.html

[8] http://lyle.smu.edu/~tspell/jaws/

[9] We also experimented with a balanced version, which contains an equal amount of sentences from the OntoNotes corpus and our scientific corpus. The results are not reported here as this version performed worse.

[10] The CoNLL score is the arithmetic mean of MUC, B^3 and CEAFE.

[11] We compute significance using the Wilcoxon signed rank test (Siegel and Castellan, 1988) at the 0.01 level unless otherwise stated.

(+9.64 and +12.62 in the GEN papers, +2.11 and +7.14 in the CL papers, absolute in CoNLL score).

IMS coreference resolver, adapted to the domain. We show the results from the expansion of the feature set in Table 8. Each of the single features is added to the version in the line above the current version. Compared to the MIXED version, adding the features to the resolver results in an increase in performance for both of the scientific disciplines. However, when adding the collocation feature to the version including the bias lists, the argumentation nouns as well as idioms, performance drops slightly. This might indicate the need for a revised collocation list where those nouns are filtered out that could potentially be coreferring, e.g. *see figure*. For the best version of the CL papers, the increase in CoNLL score, compared with the MIXED version, is +1.08; for the GEN papers it is slightly less, namely +0.22. This increase in score is promising, but the data is too small to show significance.

	GEN	CL	GEN+CL
Mixed	47.92	47.44	47.68
+ Bias Lists	48.04	47.79	47.94
+ Arg. Nouns and Idioms	**48.14**	**48.52**	**48.33**
+ Collocations	48.03	48.12	48.08

Table 8: Resolving coreferent references: CoNLL scores for the extended feature sets

However, compared with the BASELINE version, the final version (marked bold) performs significantly better and outperforms the out-of-the-box run by 36.47 % absolute on the CoNLL metric for the GEN papers and by 20.40 % for the CL papers. The results also show that, in our experiments, the effect of using domain-specific training material is larger than the effect of adding terminological features.

6.3 Resolving Associative References

IMS coreference resolver unchanged. As associative references are not annotated in the OntoNotes dataset, the only possible baseline we can use is the system trained on the scientific corpus. Average CoNLL scores were 33.52 for GEN and 32.86 for CL (33.14 overall). As expected, the performance on associative anaphora is worse than on coreferent anaphora. We have not made any changes to the resolver, so it is interesting to see that the resolver is indeed able to adjust to the new notion of reference and manages to link associative references.

We found that the resolver generally identifies very few associative references and so the most common error of the system is that it fails to recognise associative relations, particularly if the computed edit distance, one of the standard features in the coreference resolver, is very high. The easiest associative relations to detect are those which have similar surface forms. For example, the coreference resolver correctly links *RNAI* and *RNAI genes*, *the sense strand* and *the anti-sense strand* or *siRNAs* and *efficacious siRNAs*. However, for most of the associative references, the lack of easily identifiable surface markers makes the task difficult. Ironically, the system also falsely classifies many coreference links as associative, although it has this time of course been trained only on associative references. This is not surprising, given that the tasks are so similar that we are able to use a coreference resolver for the associative task in the first place.

IMS coreference resolver using semantic features. Table 9 gives the results of the extended feature set that includes the semantic features described in Section 5.2. Each of the respective semantic features shown in the table is added to the version in the line above the current version.
It can be seen that the different WordNet relations have different effects on the two scientific disciplines. For the genetics papers, the inclusion of synonyms, hyponyms and hypernyms results in the highest increase in performance (+2.02). For the computational linguistics papers, the inclusion of synonyms, hyponyms, topics and meronyms obtains the best performance (+1.19). As the effect of the features is discipline-dependent, we create two separate final feature sets for the two disciplines. The GEN version contains synonyms, hyponyms and hypernyms while the CL version contains synonyms, hyponyms, topics and meronyms. The highest increase in performance for the CL feature set (and the one resulting in the final system) was achieved by dropping topic members and hypernyms. In the final CL system, the increase in performance compared to the baseline version is +1.35. Both final versions significantly outperform the baseline.

When comparing the output of the extended system to the baseline system, it can be seen that

	GEN	CL	GEN+CL
Baseline	33.52	32.86	33.19
+ Synonyms	33.95	32.87	33.41
+ Hyponyms	34.04	32.94	33.49
+ Hypernyms	**35.54**	31.35	33.45
+ Topic members	34.61	30.61	32.61
+ Topics	34.09	32.88	33.49
+ Meronyms	33.70	**34.05**	**33.88**
+ Substance meronyms	33.57	32.40	32.99
Final version (domain-dependent)	**35.54**	**34.21**	**34.88**

Table 9: Resolving associative references: CoNLL metric scores for the extended feature sets

the resolver now links many more mentions (5.7 times more in the GEN papers, 3.8 times more in the CL papers). The reason why this does not lead to an even larger increase in performance lies in the large number of false positives. However, when looking at the data it becomes apparent that the newly created links are mostly links that potentially could have been annotated during the annotation, but are not part of the gold standard because the associative antecedent is not absolutely necessary in order to interpret the anaphor or because the entity has been linked to a different entity where the associative relation is stronger. The existence of more-or-less acceptable alternative associative links casts some doubt on using a gold standard as the sole evaluation criterion. An alternative would be to ask humans for a rating of the sensibility of the links determined by the system.

7 Conclusion and Future Work

We have presented a study of information status in two scientific disciplines as well as preliminary experiments on the resolution of both coreferent and associative anaphora in these disciplines. Our results show a marked difference in the distributions of information status categories between scientific and news text. Our corpus of 16 full-text scholarly papers annotated with information status and anaphoric links, which we plan to release soon, contains over 8000 annotated definite noun phrases. We demonstrate that the integration of domain-dependent terminological features, combined with domain-dependent training data, outperforms the unadjusted IMS system (Björkelund and Farkas, 2012) by 36.47 % absolute on the CoNLL metric for the genetics papers and by 20.40 % absolute for the computational linguistics papers. The effect of domain-dependent training material was stronger than the integration of terminological features. As far as the resolution of associative anaphora is concerned, we have shown that it is generally possible to adapt a coreference resolver to this task, and we have achieved an improvement in performance using novel semantic features. We are currently performing an inter-annotator study with two additional annotators, which will also lead to a better understanding of the relative difficulty of the categories. Furthermore, we plan to convert the coreference-annotated ACL papers by Schäfer et al. (2012) into CoNLL format and use them for training the coreference resolver. As we have annotated our corpus with information status, it might also be interesting to train a classifier on the information status categories and use its predictions to improve the performance on anaphora resolution tasks. To do so, we will create a separate corpus for testing, annotated solely with coreference and associative links.

Acknowledgements

We would like to thank Jonas Kuhn, Ulrich Heid and Arndt Riester for their valuable comments as well as the anonymous reviewers for their insightful remarks. We gratefully acknowledge financial support from the Deutsche Forschungsgemeinschaft (DFG) in the framework of project B3 and D7 of the SFB 732. This research is partially supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center (DoI/NBC) contract number D11PC20153. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

References

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

Riza T. Batista-Navarro and Sophia Ananiadou. 2011. Building a coreference-annotated corpus from the domain of biochemistry. In *Proceedings of BioNLP 2011 Workshop*, pages 83–91. Association for Computational Linguistics.

Anders Björkelund and Richárd Farkas. 2012. Data-driven multilingual coreference resolution using resolver stacking. In *Joint Conference on EMNLP and CoNLL-Shared Task*, pages 49–55. Association for Computational Linguistics.

Chen Chen and Vincent Ng. 2012. Combining the best of two worlds: A hybrid approach to multilingual coreference resolution. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 56–63, Jeju Island, Korea, July. Association for Computational Linguistics.

Herbert H. Clark. 1975. Bridging. In *Proceedings of the 1975 workshop on Theoretical issues in natural language processing*, pages 169–174. Association for Computational Linguistics.

Kevin B. Cohen, Arrick Lanfranchi, William Corvey, William A. Baumgartner Jr, Christophe Roeder, Philip V. Ogren, Martha Palmer, and Lawrence Hunter. 2010. Annotation of all coreference in biomedical text: Guideline selection and adaptation. In *Proceedings of BioTxtM 2010: 2nd workshop on building and evaluating resources for biomedical text mining*, pages 37–41.

Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, October. Association for Computational Linguistics.

Yimai Fang and Simone Teufel. 2014. A summariser based on human memory limitations and lexical competition. In *Proceedings of the EACL*. Association for Computational Linguistics. (to appear).

Eraldo Fernandes, Cícero dos Santos, and Ruy Milidiú. 2012. Latent structure perceptron with feature induction for unrestricted coreference resolution. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 41–48, Jeju Island, Korea, July. Association for Computational Linguistics.

Caroline V. Gasperin. 2009. Statistical anaphora resolution in biomedical texts. Technical Report UCAM-CL-TR-764, University of Cambridge, Computer Laboratory, December.

Anita Gojun, Ulrich Heid, Bernd Weissbach, Carola Loth, and Insa Mingers. 2012. Adapting and evaluating a generic term extraction tool. In *Proceedings of the 8th international conference on Language Resources and Evaluation (LREC)*, pages 651–656.

Marti A. Hearst. 1994. Multi-paragraph segmentation of expository text. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 9–16.

Yufang Hou, Katja Markert, and Michael Strube. 2013. Global inference for bridging anaphora resolution. In *Proceedings of NAACL-HLT*, pages 907–917.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: the 90% solution. In *Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers*, pages 57–60. Association for Computational Linguistics.

Dain Kaplan, Ryu Iida, Kikuko Nishina, and Takenobu Tokunaga. 2012. Slate – a tool for creating and maintaining annotated corpora. *Journal for Language Technology and Computational Linguistics*, pages 89–101.

Walter Kintsch and Teun A. van Dijk. 1978. Toward a model of text comprehension and production. *Psychological Review*, 85(5):363–394.

Katja Markert, Yufang Hou, and Michael Strube. 2012. Collective classification for fine-grained information status. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 795–804. Association for Computational Linguistics.

Shachar Mirkin, Ido Dagan, and Sebastian Padó. 2010. Assessing the role of discourse references in entailment inference. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL 2010, pages 1209–1219. Association for Computational Linguistics.

Vincent Ng. 2010. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1396–1411. Association for Computational Linguistics.

Malvina Nissim, Shipra Dingare, Jean Carletta, and Mark Steedman. 2004. An annotation scheme for information status in dialogue. *LREC 2004*.

Massimo Poesio and Renata Vieira. 1998. A corpus-based investigation of definite description use. *Computational linguistics*, 24(2):183–216.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Proceedings of the Joint Conference on EMNLP and CoNLL: Shared Task*, pages 1–40.

Ellen F. Prince. 1981. Toward a taxonomy of given-new information. In *Radical Pragmatics*, pages 223–55. Academic Press.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. A multi-pass sieve for coreference resolution. In *Proceedings of EMNLP 2010*.

Arndt Riester, David Lorenz, and Nina Seemann. 2010. A recursive annotation scheme for referential information status. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)*, pages 717–722.

Ulrich Schäfer, Christian Spurk, and Jörg Steffen. 2012. A fully coreference-annotated corpus of scholarly papers from the acl anthology. In *Proceedings of the 24th International Conference on Computational Linguistics. International Conference on Computational Linguistics (COLING-2012), December 10-14, Mumbai, India*, pages 1059–1070.

Sidney Siegel and N. John Jr. Castellan. 1988. *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill, Berkeley, CA, 2nd edition.

Simone Teufel. 2010. *The Structure of Scientific Articles: Applications to Citation Indexing and Summarization*. CSLI Publications.

Rebecca Watson, Judita Preiss, and Ted Briscoe. 2003. The contribution of domain-independent robust pronominal anaphora resolution to open-domain question-answering. In *Proceedings of the Symposium on Reference Resolution and its Applications to Question Answering and Summarization. Venice, Italy June*, pages 23–25.

Modelling Irony in Twitter

Francesco Barbieri
Pompeu Fabra University
Barcelona, Spain
francesco.barbieri@upf.edu

Horacio Saggion
Pompeu Fabra University
Barcelona, Spain
horacio.saggion@upf.edu

Abstract

Computational creativity is one of the central research topics of Artificial Intelligence and Natural Language Processing today. Irony, a creative use of language, has received very little attention from the computational linguistics research point of view. In this study we investigate the automatic detection of irony casting it as a classification problem. We propose a model capable of detecting irony in the social network Twitter. In cross-domain classification experiments our model based on lexical features outperforms a word-based baseline previously used in opinion mining and achieves state-of-the-art performance. Our features are simple to implement making the approach easily replicable.

1 Introduction

Irony, a creative use of language, has received very little attention from the computational linguistics research point of view. It is however considered an important aspect of language which deserves special attention given its relevance in fields such as sentiment analysis and opinion mining (Pang and Lee, 2008). Irony detection appears as a difficult problem since ironic statements are used to express the contrary of what is being said (Quintilien and Butler, 1953), therefore being a tough nut to crack by current systems. Being a creative form of language, there is no consensual agreement in the literature on how verbal irony should be defined. Only recently irony detection has been approached from a computational perspective. Reyes et al. (2013) cast the problem as one of classification training machine learning algorithms to sepatare ironic from non-ironic statements. In a similar vein, we propose and evaluate a new model to detect irony, using seven sets of lexical features, most of them based on our intuitions about "unexpectedness", a key component of ironic statements. Indeed, Lucariello (1994) claims that irony is strictly connected to surprise, showing that unexpectedness is the feature most related to situational ironies.

In this paper we reduce the complexity of the problem by studying irony detection in the micro-blogging service Twitter[1] that allows users to send and read text messages (shorter than 140 characters) called tweets.

We do not adopt any formal definition of irony, instead we rely on a dataset created for the study of irony detection which allows us to compare our findings with recent state-of-the-art approaches (Reyes et al., 2013).

The contributions of this paper are as follows:

- a novel set of linguistically motivated, easy-to-compute features
- a comparison of our model with the state-of-the-art; and
- a novel set of experiments to demonstrate cross-domain adaptation.

The paper will show that our model outperforms a baseline, achieves state-of-the-art performance, and can be applied to different domains.

The rest of the paper is organised as follows: in the next Section we describe related work. In Section 3 we described the corpus and text processing tools used and in Section 4 we present our approach to tackle the irony detection problem. Section 5 describes the experiments while Section 6 interprets the results. Finally we close the paper in Section 7 with conclusions and future work.

[1] https://twitter.com/

Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 56–64,
Gothenburg, Sweden, April 26-30 2014.

2 Related Work

Verbal irony has been defined in several ways over the years but there is no consensual agreement on the definition. The standard definition is considered “saying the opposite of what you mean” (Quintilien and Butler, 1953) where the opposition of literal and intended meanings is very clear. Grice (1975) believes that irony is a rhetorical figure that violates the maxim of quality: “Do not say what you believe to be false”. Irony is also defined (Giora, 1995) as any form of negation with no negation markers (as most of the ironic utterances are affirmative, and ironic speakers use indirect negation). Wilson and Sperber (2002) defined it as echoic utterance that shows a negative aspect of someone’s else opinion. For example if someone states “the weather will be great tomorrow” and the following day it rains, someone with ironic intents may repeat the sentence “the weather will be great tomorrow” in order to show the statements was incorrect. Finally irony has been defined as form of pretence by Utsumi (2000) and Veale and Hao (2010b). Veale states that “ironic speakers usually craft their utterances in spite of what has just happened, not because of it. The pretence alludes to, or echoes, an expectation that has been violated”.

Past computational approaches to irony detection are scarce. Carvalho et. al (2009) created an automatic system for detecting irony relying on emoticons and special punctuation. They focused on detection of ironic style in newspaper articles. Veale and Hao (2010a) proposed an algorithm for separating ironic from non-ironic similes, detecting common terms used in this ironic comparison. Reyes et. al (2013) have recently proposed a model to detect irony in Twitter, which is based on four groups of features: signatures, unexpectedness, style, and emotional scenarios. Their classification results support the idea that textual features can capture patterns used by people to convey irony. Among the proposed features, *skip-grams* (part of the style group) which captures word sequences that contain (or skip over) arbitrary gaps, seems to be the best one.

There are also a few computational model that detect sarcasm ((Davidov et al., 2010); (González-Ibáñez et al., 2011); (Liebrecht et al., 2013)) on Twitter and Amazon, but even if one may argue that sarcasm and irony are the same linguistic phenomena, the latter is more similar to mocking or making jokes (sometimes about ourselves) in a sharp and non-offensive manner. On the other hand, sarcasm is a meaner form of irony as it tends to be offensive and directed towards other people (or products like in Amazon reviews). Textual examples of sarcasm lack the sharp tone of an aggressive speaker, so for textual purposes we think irony and sarcasm should be considered as different phenomena and studied separately (Reyes et al., 2013).

3 Data and Text Processing

The dataset used for the experiments reported in this paper has been prepared by Reyes et al. (2013). It is a corpus of 40.000 tweets equally divided into four different topics: *Irony*, *Education*, *Humour*, and *Politics* where the last three topics are considered non-ironic. The tweets were automatically selected by looking at Twitter hashtags (#irony, #education, #humour, and #politics) added by users in order to link their contribution to a particular subject and community. The hashtags are removed from the tweets for the experiments. According to Reyes et. al (2013), these hashtags were selected for three main reasons: (i) to avoid manual selection of tweets, (ii) to allow irony analysis beyond literary uses, and because (iii) irony hashtag may “reflect a tacit belief about what constitutes irony.”

Another corpora is employed in our approach to measure the frequency of word usage. We adopted the Second Release of the American National Corpus Frequency Data[2] (Ide and Suderman, 2004), which provides the number of occurrences of a word in the written and spoken ANC. From now on, we will mean with “frequency of a term” the absolute frequency the term has in the ANC.

3.1 Text Processing

In order to process the tweets we use the freely available *vinhkhuc* Twitter Tokenizer[3] which allows us to recognise words in each tweet. To part-of-speech tag the words, we rely on the Rita WordNet API (Howe, 2009) that associates to a word with its most frequently used part of speech. We also adopted the Java API for WordNet Searching

[2] The American National Corpus (http://www.anc.org/) is, as we read in the web site, a massive electronic collection of American English words (15 million)

[3] https://github.com/vinhkhuc/Twitter-Tokenizer/blob/master/src/Twokenizer.java

(Spell, 2009) to perform some operation on WordNet synsets. It is worth noting that although our approach to text processing is rather superficial for the moment, other tools are available to perform deeper tweet linguistic analysis (Bontcheva et al., 2013; Derczynski et al., 2013).

4 Methodology

We approach the detection of irony as a classification problem applying supervised machine learning methods to the Twitter corpus described in Section 3. When choosing the classifiers we had avoided those requiring features to be independent (e.g. Naive Bayes) as some of our features are not. Since we approach the problem as a binary decision (deciding if a tweet is ironic or not) we picked two tree-based classifiers: Random Forest and Decision tree (the latter allows us to compare our findings directly to Reyes et. al (2013)). We use the implementations available in the Weka toolkit (Witten and Frank, 2005).

To represent each tweet we use six groups of features. Some of them are designed to detect imbalance and unexpectedness, others to detect common patterns in the structure of the ironic tweets (like type of punctuation, length, emoticons). Below is an overview of the group of features in our model:

- Frequency *(gap between rare and common words)*
- Written-Spoken *(written-spoken style uses)*
- Intensity *(intensity of adverbs and adjectives)*
- Structure *(length, punctuation, emoticons)*
- Sentiments *(gap between positive and negative terms)*
- Synonyms *(common vs. rare synonyms use)*
- Ambiguity *(measure of possible ambiguities)*

In our knowledge Frequency, Written Spoken, Intensity and Synonyms groups have not been used before in similar studies. The other groups have been used already (for example by Carvalho et. al (2009) or Reyes et al. (2013)) yet our implementation is different in most of the cases.

In the following sections we describe the theoretical motivations behind the features and how them have been implemented.

4.1 Frequency

As said previously unexpectedness can be a signal of irony and in this first group of features we try to detect it. We explore the frequency imbalance between words, i.e. register inconsistencies between terms of the same tweet. The idea is that the use of many words commonly used in English (i.e. high frequency in ANC) and only a few terms rarely used in English (i.e. low frequency in ANC) in the same sentence creates imbalance that may cause unexpectedness, since within a single tweet only one kind of register is expected. We are able to explore this aspect using the ANC Frequency Data corpus.

Three features belong to this group: **frequency mean**, **rarest word**, **frequency gap**. The first one is the arithmetic average of all the frequencies of the words in a tweet, and it is used to detect the *frequency style* of a tweet. The second one, **rarest word**, is the frequency value of the rarest word, designed to capture the word that may create imbalance. The assumption is that very rare words may be a sign of irony. The third one is the absolute difference between the first two and it is used to measure the imbalance between them, and capture a possible intention of surprise. We have verified that the mean of this gap in each tweet of the irony corpus is higher than in the other corpora.

4.2 Written-Spoken

Twitter is composed of written text, but an informal spoken English style is often used. We designed this set of features to explore the unexpectedness created by using spoken style words in a mainly written style tweet or vice versa (formal words usually adopted in written text employed in a spoken style context). We can analyse this aspect with ANC written and spoken, as we can see using this corpora whether a word is more often used in written or spoken English. There are three features in this group: **written mean**, **spoken mean**, **written spoken gap**. The first and second ones are the means of the frequency values, respectively, in written and spoken ANC corpora of all the words in the tweet. The third one, **written spoken gap**, is the absolute value of the difference between the first two, designed to see if ironic writers use both styles (creating imbalance) or only one of them. A low difference between written and spoken styles means that both styles are used.

4.3 Structure

With this group of features we want to study the structure of the tweet: if it is long or short (length), if it contains long or short words (mean of word length), and also what kind of punctuation is used (exclamation marks, emoticons, etc.). This is a powerful feature, as ironic tweets in our corpora present specific structures: for example they are often longer than the tweets in the other corpora, they contain certain kind of punctuation and they use only specific emoticons. This group includes several features that we describe below.

The **length** feature consists of the number of characters that compose the tweet, **n. words** is the number of words, and **words length mean** is the mean of the words length. Moreover, we use the number of verbs, nouns, adjectives and adverbs as features, naming them **n. verbs**, **n. nouns**, **n. adjectives** and **n. adverbs**. With these last four features we also computed the ratio of each part of speech to the number of words in the tweet; we called them **verb ratio**, **noun ratio**, **adjective ratio**, and **adverb ratio**. All these features have the purpose of capturing the style of the writer. Some of them seem to be significant; for example the average length of an ironic tweet is 94.8 characters and the average length of education, humour, and politics tweets are respectively 82.0, 86.6, and 86.5. The words used in the irony corpus are usually shorter than in the other corpora, but they amount to more.

The **punctuation** feature is the sum of the number of commas, full stops, ellipsis and exclamation that a tweet presents. We also added a feature called **laughing** which is the sum of all the internet laughs, denoted with *hahah*, *lol*, *rofl*, and *lmao* that we consider as a new form of punctuation: instead of using many exclamation marks internet users may use the sequence *lol* (i.e. laughing out loud) or just type *hahaha*. As the previous features, punctuation and laughing occur more frequently in the ironic tweets than in the other topics.

The **emoticon** feature is the sum of the emoticons *:)*, *:D*, *:(* and *;)* in a tweet. This feature works well in the humour corpus because is the one that presents a very different number of them, it has four times more emoticons than the other corpora. The ironic corpus is the one with the least emoticons (there are only 360 emoticons in the Irony corpus, while in Humour, Education, and Politics tweets they are 2065, 492, 397 respectively). In the light of these statistics we can argue that ironic authors avoid emoticons and leave words to be the central thing: the audience has to understand the irony without explicit signs, like emoticons. Another detail is the number of winks *;)*. In the irony corpus one in every five emoticon is a wink, whereas in the Humour, Education and Politics corpora the number of winks are 1 in every 30, 22 and 18 respectively. Even if the wink is not a usual emoticon, ironic authors use it more often because they mean *something else* when writing their tweets, and a wink is used to suggest that something is hidden behind the words.

4.4 Intensity

A technique ironic authors may employ is saying the opposite of what they mean (Quintilien and Butler, 1953) using adjectives and adverbs to, for example, describe something very big to denote something very small (e.g. saying "Do we hike that *tiny* hill now?" before going on top of a very high mountain). In order to produce an ironic effect some authors might use an expression which is antonymic to what they are trying to describe, we believe that in the case the word being an adjective or adverb its intensity (more or less exaggerated) may well play a role in producing the intended effect. We adopted the intensity scores of Potts (2011) who uses naturally occurring metadata (star ratings on service and product reviews) to construct adjectives and adverbs scales. An example of adjective scale (and relative scores in brackets) could be the following: horrible (-1.9) → bad (-1.1) → good (0.2) → nice (0.3) → great (0.8).

With these scores we evaluate four features for adjective intensity and four for adverb intensity (implemented in the same way): **adj (adv) tot**, **adj (adv) mean**, **adj (adv) max**, and **adj (adv) gap**. The sum of the AdjScale scores of all the adjectives in the tweet is called **adj tot**. **adj mean** is **adj tot** divided by the number of adjectives in the tweet. The maximum AdjScale score within a single tweet is **adj max**. Finally, **adj gap** is the difference between **adj max** and **adj mean**, designed to see "how much" the most intense adjective is out of context.

4.5 Synonyms

Ironic authors send two messages to the audience at the same time, the literal and the figurative one (Veale, 2004). It follows that the choice of a term (rather than one of its synonyms) is very important in order to send the second, not obvious, message. For example if the sky is grey and it is about to rain, someone with ironic intents may say "sublime weather today", choosing *sublime* over many different, more common, synonyms (like nice, good, very good and so on, that according to ANC are more used in English) to advise the listener that the literal meaning may not be the only meaning present. A listener will grasp this hidden information when he asks himself why a rare word like *sublime* was used in that context where other more common synonyms were available to express the same *literal* meaning.

For each word of a tweet we get its synonyms with WordNet (Miller, 1995), then we calculate their ANC frequencies and sort them into a decreasing ranked list (the actual word is part of this ranking as well). We use these rankings to define the four features which belong to this group. The first one is **syno lower** which is the number of synonyms of the word w_i with frequency lower than the frequency of w_i. It is defined as in Equation 1:

$$sl_{w_i} = |syn_{i,k} \; : \; f(syn_{i,k}) < f(w_i)| \tag{1}$$

where $syn_{i,k}$ is the synonym of w_i with rank k, and $f(x)$ the ANC frequency of x. Then we also defined **syno lower mean** as mean of sl_{w_i} (i.e. the arithmetic average of sl_{w_i} over all the words of a tweet).

We also designed two more features: **syno lower gap** and **syno greater gap**, but to define them we need two more parameters. The first one is *word lowest syno* that is the maximum sl_{w_i} in a tweet. It is formally defined as:

$$wls_t = \max_{w_i}\{|syn_{i,k} \; : \; f(syn_{i,k}) < f(w_i)|\} \tag{2}$$

The second one is *word greatest syno* defined as:

$$wgs_t = \max_{w_i}\{|syn_{i,k} \; : \; f(syn_{i,k}) > f(w_i)|\} \tag{3}$$

We are now able to describe **syno lower gap** which detects the imbalance that creates a common synonym in a context of rare synonyms. It is the difference between *word lowest syno* and **syno lower mean**. Finally, we detect the gap of very rare synonyms in a context of common ones with **syno greater gap**. It is the difference between *word greatest syno* and *syno greater mean*, where *syno greater mean* is the following:

$$sgm_t = \frac{|syn_{i,k} \; : \; f(syn_{i,k}) > f(w_i)|}{n.\,words\,of\,t} \tag{4}$$

The arithmetic averages of **syno greater gap** and of **syno lower gap** in the irony corpus are higher than in the other corpora, suggesting that a very common (or very rare) synonym is often used out of context i.e. a very rare synonym when most of the words are common (have a high rank in our model) and vice versa.

4.6 Ambiguity

Another interesting aspect of irony is ambiguity. We noticed that the arithmetic average of the number of WordNet synsets in the irony corpus is greater than in all the other corpora; this indicates that ironic tweets presents words with more meanings. Our assumption is that if a word has many meanings the possibility of "saying something else" with this word is higher than in a term that has only a few meanings, then higher possibility of sending more then one message (literal and intended) at the same time.

There are three features that aim to capture these aspects: **synset mean**, **max synset**, and **synset gap**. The first one is the mean of the number of synsets of each word of the tweet, to see if words with many meanings are often used in the tweet. The second one is the greatest number of synsets that a single word has; we consider this word the one with the highest possibility of being used ironically (as multiple meanings are available to say different things). In addition, we calculate **synset gap** as the difference between the number of synsets of this word (**max synset**) and the average number of synsets (**synset mean**), assuming that if this gap is high the author may have used that inconsistent word intentionally.

4.7 Sentiments

We think that sign of irony could also be found using sentiment analysis. The SentiWordNet sentiment lexicon (Esuli and Sebastiani, 2006) assigns to each synset of WordNet sentiment scores of positivity and negativity. We used these scores to examine what kind of sentiments characterises irony. We explore ironic sentiments with two different views: the first one is the simple analysis of sentiments (to identify the main sentiment that arises from ironic tweets) and the second one concerns sentiment imbalances between words, de-

	Training Set								
	Education			**Humour**			**Politics**		
Test set	P	R	F1	P	R	F1	P	R	F1
Education	.85/.73	.84/.73	.84/.73	.57/.61	.53/.61	.46/**.61**	.61/.67	.56/.67	.51/**.67**
Humour	.64/.62	.51/.62	.58/**.62**	.85/.75	.85/.75	.85/.75	.65/.61	.59/.61	.55/**.60**
Politics	.61/.67	.58/.67	.55/**.67**	.55/.61	.60/.60	.56/**.60**	.87/.75	.87/.75	.87/.75

Table 1: Precision, Recall and F-Measure of each topic combination for word based algorithm and our algorithm in the form "Word Based / Ours". Decision Tree has been used as classifier for both algorithms. We marked in **bold** the results that, according to the t-test, are significantly better.

signed to explore unexpectedness from a sentiment prospective.

There are six features in the Sentiments group. The first one is named **positive sum** and it is the sum of all the positive scores in a tweet, the second one is **negative sum**, defined as sum of all the negative scores. The arithmetic average of the previous ones is another feature, named **positive negative mean**, designed to reveal the sentiment that better describe the whole tweet. Moreover, there is **positive-negative gap** that is the difference between the first two features, as we wanted also to detect the positive/negative imbalance within the same tweet.

The imbalance may be created using only one single very positive (or negative) word in the tweet, and the previous features will not be able to detect it, thus we needed to add two more. For this purpose the model includes **positive single gap** defined as the difference between most positive word and the mean of all the sentiment scores of all the words of the tweet and **negative single gap** defined in the same way, but with the most negative one.

4.8 Bag of Words Baseline

Based on previous work on sentiment analysis and opinon classification (see (Pang et al., 2002; Dave et al., 2003) for example) we also investigate the value of using bag of words representations for irony classification. In this case, each tweet is represented as a set of word features. Because of the brevity of tweets, we are only considering presence/absence of terms instead of frequency-based representations based on $tf * idf$.

5 Experiments and Results

In order to carry out experimentation and to be able to compare our approach to that of (Reyes et al., 2013) we use three datasets derived from the corpus in Section 3. Irony vs Education, Irony vs Humour and Irony vs Politics. Each topic combination was balanced with 10.000 ironic and 10.000 of non-ironic examples. The task at hand it to train a classifier to identify ironic and non-ironic tweets.

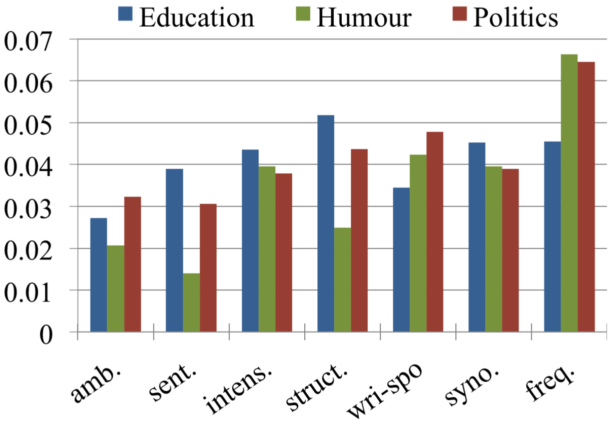


Figure 1: Information gain value of each group (mean of the features belonged to each group) over the three balanced corpus.

We perform two types of experiments:

- we run in each of the datasets a 10-fold cross-validation classification;
- across datasets, we train the classifier in one dataset and apply it to the other two datasets. To perform these experiments, we create three balanced datasets containing each one third of the original 10.000 ironic tweets (so that the datasets are disjoint) and one third of the original domain tweets.

The experimental framework is executed for the word-based baseline model and our model. In Table 1 we present precision, recall, and F-measure

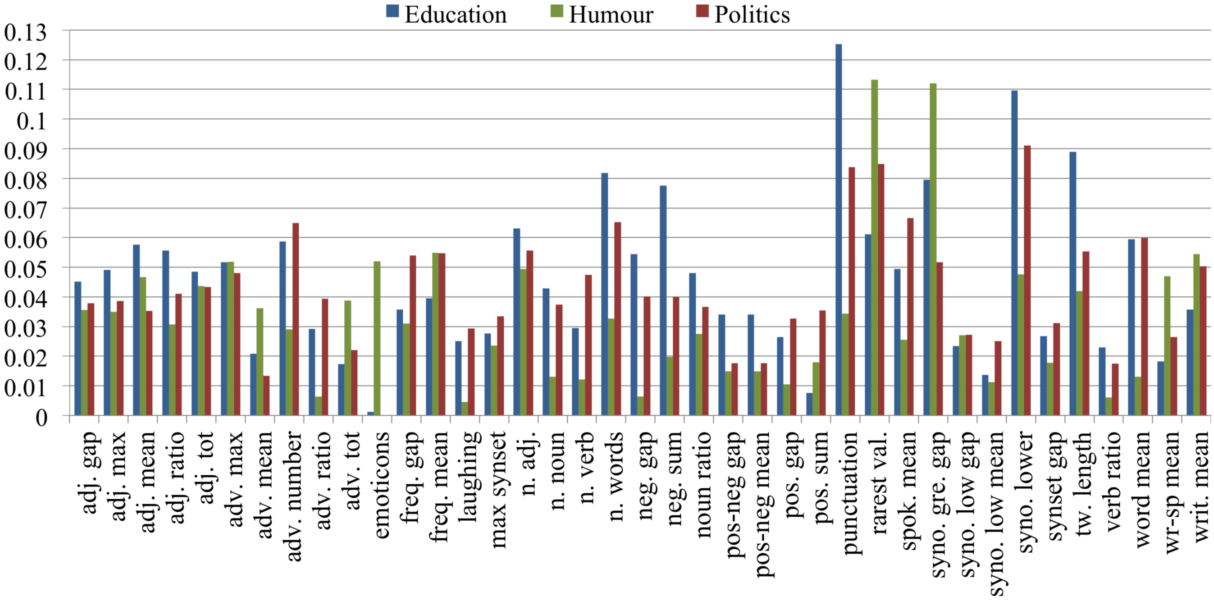


Figure 2: Information gain of each feature of the model. Irony corpus is compared to Education, Humor, and Politics corpora. High values of information gain help to better discriminate ironic from non-ironic tweets.

figures for the different runs of the experiments. Table 3 shows precision, recall, and F-measure figures for our approach compared to (Reyes et al., 2013). Table 2 compares two different algorithms: Decision Tree and Random Forest using our model.

In order to have a clear understanding about the contribution of each set of features in our model, we also studied the behaviour of information gain in each dataset. We compute information gain experiments over the three balanced corpora and present the results in Figure 1. The graphic shows the mean information gain for each group of features. We also report in Figure 2 the information gain of each single feature, where one can understand if a feature will be important to distinguish ironic from non-ironic tweets.

6 Discussion

The results obtained with the bag-of-words baseline seem to indicate that this approach is working as a topic-based classifier and not as an irony detection procedure. Indeed, within each domain using a 10 fold cross-validation setting, the bag-of-words approach seems to overtake our model. However, a clear picture emerges when a cross-domain experiment is performed. In a setting where different topics are used for training and testing our model performs significantly better than the baseline. t-tests were run for each experiment and differences between baseline and our model were observed for each cross-domain condition (with a 99% confidence level). This could be an indication that our model is more able to capture ironic style disregarding domain.

Analysing the data on Figure 2, we observe that features which are more discriminative of ironic style are **rarest value**, **synonym lower**, **synonym greater gap**, and **punctuation**, suggesting that Frequency, Structure and choice of the Synonym are important aspects to consider for irony detection in tweets (this latter statement can be appreciated in Figure 1 as well). Note, however, that there is a topic or theme effect since features behave differently depending on the dataset used: the Humour corpus seems to be the least consistent. For instance **punctuation** well distinguishes ironic from educational tweets, but behaves poorly in the Humour corpus. This imbalance may cause issues in a not controlled environment (e.g. no preselected topics, only random generic tweets). In spite of this, information gain values are fairly high with four features having information gain values over 0.1. Finding features that are significant for any non-ironic topic is hard, this is why our system includes several feature sets: they aim to distinguish irony from as many different topics as possible.

	Training Set								
	Education			**Humour**			**Politics**		
Test set	P	R	F1	P	R	F1	P	R	F1
Education	.78/.73	.78/.73	**.78**/.73	.65/.61	.63/.61	.62/.61	.71/.67	.71/.67	.70/.67
Humour	.64/.62	.61/.62	.60/.62	.80/.75	.80/.75	**.80**/.75	.64/.61	.62/.61	.60/.60
Politics	.71/.67	.70/.67	.69/.67	.63/.61	.51/.60	.59/.60	.79/.75	.79/.75	**.79**/.75

Table 2: Precision, Recall and F-Measure for each topic combination of our model when Decision Tree and Random Forest are used. Data are in the format "Random Forest / Decision Tree". We marked in **bold** the F-Measures that are better.

	Education			**Humour**			**Politics**		
Model	P	R	F1	P	R	F1	P	R	F1
Reyes et. al	.76	.66	.70	.78	.74	**.76**	.75	.71	.73
Our model	.73	.73	**.73**	.75	.75	.75	.75	.75	**.75**

Table 3: Precision, Recall, and F-Measure over the three corpora Education, Humour, and Politics. Both our and Reyes et al. results are shown; the classifier used is Decision Tree for both models. We marked in **bold** the F-Measures that are better compared to the other model.

With respect to results for two different classifiers trained with our model (Random Forest (RF) and Decision Trees (DT)) we observe that (see Table 2) RF is better in cross-validation but across-domains both algorithms are comparable.

Turning now to the state of the art we compare our approach to (Reyes et al., 2013), the numbers presented in Table 3 seem to indicate that (i) our approach is more balanced in terms of precision and recall and that (ii) our approach performs slightly better in terms of F-Measure in two out of three domains.

7 Conclusion and Future Work

In this article we have proposed a novel linguistically motivated set of features to detect irony in the social network Twitter. The features take into account frequency, written/spoken differences, sentiments, ambiguity, intensity, synonymy and structure. We have designed many of them to be able to model "unexpectedness", a key characteristic of irony.

We have performed controlled experiments with an available corpus of ironic and non-ironic tweets using classifiers trained with bag-of-words features and with our irony specific features. We have shown that our model performs better than a bag-of-words approach across-domains. We have also shown that our model achieves state-of-the-art performance.

There is however much space for improvements. The ambiguity aspect is still weak in this research, and it needs to be improved. Also experiments adopting different corpora (Filatova, 2012) and different negative topics may be useful in order to explore the system behaviour in a real situation. Finally, we have relied on very basic tools for linguistic analysis of the tweets, so in the near future we intend to incorporate better linguistic processors. A final aspect we want to investigate is the use of n-grams from huge collections to model "unexpected" word usage.

Acknowledgments

We are greatful to three anonymous reviewers for their comments and suggestions that help improve our paper. The research described in this paper is partially funded by fellowship RYC-2009-04291 from Programa Ramón y Cajal 2009 and project number TIN2012-38584-C06-03 (SKATER-UPF-TALN) from Ministerio de Economía y Competitividad, Secretaría de Estado de Investigación, Desarrollo e Innovacíon, Spain.

References

Kalina Bontcheva, Leon Derczynski, Adam Funk, Mark A. Greenwood, Diana Maynard, and Niraj Aswani. 2013. TwitIE: An Open-Source Information Extraction Pipeline for Microblog Text. In *Proceedings of Recent Advances in Natural Language Processing Conferemce*.

Paula Carvalho, Luís Sarmento, Mário J Silva, and

Eugénio de Oliveira. 2009. Clues for detecting irony in user-generated contents: oh...!! it's so easy;-). In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pages 53–56. ACM.

Kushal Dave, Steve Lawrence, and David M. Pennock. 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th International Conference on World Wide Web*, WWW '03, pages 519–528, New York, NY, USA. ACM.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 107–116. Association for Computational Linguistics.

Leon Derczynski, Alan Ritter, Sam Clark, and Kalina Bontcheva. 2013. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*.

Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of Language Resources and Evaluation Conference*, volume 6, pages 417–422.

Elena Filatova. 2012. Irony and sarcasm: Corpus generation and analysis using crowdsourcing. In *Proceedings of Language Resources and Evaluation Conference*, pages 392–398.

Rachel Giora. 1995. On irony and negation. *Discourse processes*, 19(2):239–264.

Roberto González-Ibáñez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying sarcasm in twitter: A closer look. In *ACL (Short Papers)*, pages 581–586. Citeseer.

H Paul Grice. 1975. Logic and conversation. *1975*, pages 41–58.

Daniel C Howe. 2009. Rita wordnet. java based api to access wordnet.

Nancy Ide and Keith Suderman. 2004. The American National Corpus First Release. In *Proceedings of the Language Resources and Evaluation Conference*.

Christine Liebrecht, Florian Kunneman, and Antal van den Bosch. 2013. The perfect solution for detecting sarcasm in tweets# not. *WASSA 2013*, page 29.

Joan Lucariello. 1994. Situational irony: A concept of events gone awry. *Journal of Experimental Psychology: General*, 123(2):129.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Bo Pang and Lillian Lee. 2008. Opinion Mining and Sentiment Analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 79–86, Stroudsburg, PA, USA. Association for Computational Linguistics.

Christopher Potts. 2011. Developing adjective scales from user-supplied textual metadata. *NSF Workshop on Restructuring Adjectives in WordNet. Arlington,VA.*

Quintilien and Harold Edgeworth Butler. 1953. *The Institutio Oratoria of Quintilian. With an English Translation by HE Butler.* W. Heinemann.

Antonio Reyes, Paolo Rosso, and Tony Veale. 2013. A multidimensional approach for detecting irony in twitter. *Language Resources and Evaluation*, pages 1–30.

Brett Spell. 2009. Java api for wordnet searching (jaws).

Akira Utsumi. 2000. Verbal irony as implicit display of ironic environment: Distinguishing ironic utterances from nonirony. *Journal of Pragmatics*, 32(12):1777–1806.

Tony Veale and Yanfen Hao. 2010a. Detecting ironic intent in creative comparisons. In *ECAI*, volume 215, pages 765–770.

Tony Veale and Yanfen Hao. 2010b. An ironic fist in a velvet glove: Creative mis-representation in the construction of ironic similes. *Minds and Machines*, 20(4):635–650.

Tony Veale. 2004. The challenge of creative information retrieval. In *Computational Linguistics and Intelligent Text Processing*, pages 457–467. Springer.

Deirdre Wilson and Dan Sperber. 2002. Relevance theory. *Handbook of pragmatics*.

Ian H Witten and Eibe Frank. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Multi-class Animacy Classification with Semantic Features

Johannes Bjerva
Center for Language and Cognition Groningen
University of Groningen
The Netherlands
j.bjerva@rug.nl

Abstract

Animacy is the semantic property of nouns denoting whether an entity can act, or is perceived as acting, of its own will. This property is marked grammatically in various languages, albeit rarely in English. It has recently been highlighted as a relevant property for NLP applications such as parsing and anaphora resolution. In order for animacy to be used in conjunction with other semantic features for such applications, appropriate data is necessary. However, the few corpora which do contain animacy annotation, rarely contain much other semantic information. The addition of such an annotation layer to a corpus already containing deep semantic annotation should therefore be of particular interest.

The work presented in this paper contains three main contributions. Firstly, we improve upon the state of the art in multi-class animacy classification. Secondly, we use this classifier to contribute to the annotation of an openly available corpus containing deep semantic annotation. Finally, we provide source code, as well as trained models and scripts needed to reproduce the results presented in this paper, or aid in annotation of other texts.[1]

1 Introduction

Animacy is the semantic property of nouns denoting whether, or to what extent, the referent of that noun is alive, human-like or even cognitively sophisticated. Several ways of characterising the animacy of such referents have been proposed in the literature, the most basic distinction being between animate and inanimate entities. In such a binary scheme, examples of animate nouns might include *author* and *dog*, while examples of inanimate nouns might include *table* and *rock*. More elaborate schemes tend to represent a hierarchy or continuum typically ranging from HUMAN → NON-HUMAN → INANIMATE (cf. Comrie (1989)), with other categories in between.

In various languages, animacy affects linguistic phenomena such as case marking and argument realization. Furthermore, hierarchical restrictions are often imposed by animacy, e.g. with subjects tending to be higher in an animacy hierarchy than objects (Dahl and Fraurud, 1996). Even though animacy is rarely overtly marked in English, it still influences the choice of certain grammatical structures, such as the choice of relative pronouns (e.g. *who* vs. *which*).

The aims of this work are as follows: *(i)* to improve upon the state of the art in multi-class animacy classification by comparing and evaluating different classifiers and features for this task, *(ii)* to investigate whether a corpus of spoken language containing animacy annotation can be used as a basis to annotate animacy in a corpus of written language, *(iii)* to use the resulting classifier as part of the toolchain used to annotate a corpus containing deep semantic annotation.

The remainder of this paper is organized as follows: In Section 2 we go through the relevance of animacy for Natural Language Processing (NLP) and describe some corpora which contain animacy annotation. Previous attempts and approaches to animacy classification are portrayed in Section 3. Section 4 contains an overview of the data used in this study, as well as details regarding the manual annotation of animacy carried out as part of this work. The methods employed and the results obtained are presented in Sections 5 and 6. The discussion is given in Section 7. Finally, Section 8 contains conclusions and some suggestions for future work in multi-class animacy classification.

[1] https://github.com/bjerva/animacy

Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 65–75,
Gothenburg, Sweden, April 26-30 2014. ©2014 Association for Computational Linguistics

2 Background

2.1 Relevance of animacy for NLP

Although seemingly overlooked in the past, animacy has recently been shown to be an important feature for NLP. Øvrelid & Nivre (2007) found that the accuracy of a dependency parser for Swedish could be improved by incorporating a binary animacy distinction. Other work has highlighted animacy as relevant for anaphora and coreference resolution (Orăsan and Evans, 2007; Lee et al., 2013) and verb argument disambiguation (Dell'Orletta et al., 2005).

Furthermore, in English, the choices for dative alternation (Bresnan et al., 2007), between genitive constructions (Stefanowitsch, 2003), and between active and passive voice (Rosenbach, 2008) are also affected by the animacy of their constituent nouns. With this in mind, Zaenen et al. (2004) suggest that animacy, for languages such as English, is not a matter of grammatical and ungrammatical sentences, but rather of sentences being more and less felicitous. This highlights annotation of animacy as potentially particularly useful for applications such as Natural Language Generation.

In spite of this, animacy appears to be rarely annotated in corpora, and thus also rather rarely used in tools and algorithms for NLP (although some recent efforts do exist, cf. Moore et al. (2013)). Furthermore, the few corpora that do include animacy in their annotation do not contain much other semantic annotation, making them less interesting for computational semanticists.

2.2 Annotation of animacy

Resources annotated with animacy are few and far between. One such resource is the MC160 dataset which has recently been labelled for binary animacy (Moore et al., 2013). The distinction between animate and inanimate was based on whether or not an entity could "move under its own will". Although interesting, the size of this data set (approximately 8,000 annotated nouns) limits its usefulness, particularly with the methods used in this paper.

Talbanken05 is a corpus of Swedish spoken language which includes a type of animacy annotation (Nivre et al., 2006). However, this annotation is better described as a distinction between human and non-human, than between animate and inanimate (Øvrelid, 2009). Although the work in this paper focusses on English, a potential application of this corpus is discussed at the end of this paper (see Section 8).

The NXT Switchboard corpus represents a larger and more interesting resource for our purposes (Calhoun et al., 2010). This spoken language corpus contains high quality manual annotation of animacy for nearly 200,000 noun phrases (Zaenen et al., 2004). Furthermore, the annotation is fairly fine-grained, as a total of ten animacy categories are used (see Table 1), with a few additional tags for mixed animacy and cases in which annotators were uncertain. This scheme can be arranged hierarchically, so that the classes Concrete, Non-concrete, Place and Time are grouped as inanimate, while the remaining classes are grouped as animate. The availability of this data allows us to easily exploit the annotation for a supervised learning approach (see Section 5).

3 Related work

In this section we will give an overview of previous work in animacy classification, some of which has inspired the approach presented in this paper.

3.1 Exploiting corpus frequencies

A binary animacy classifier which uses syntactic and morphological features has been previously developed for Norwegian and Swedish (Øvrelid, 2005; Øvrelid, 2006; Øvrelid, 2009). The features used are based on frequency counts from the dependency-parsed Talbanken05 corpus. These frequencies are counted per noun lemma, meaning that this classifier is not context sensitive. In other words, cases of e.g. polysemy where *head* is inanimate in the sense of *human head*, but animate in the sense of *head of an organization*, are likely to be problematic. Intuitively, by taking context or semantically motivated features into account, such cases ought to be resolved quite trivially.

This classifier performs well, as it reaches an accuracy for 96.8% for nouns, as compared to a baseline of 90.5% when always picking the most common class (Øvrelid, 2009). Furthermore, it is shown that including the binary distinction from this classifier as a feature in dependency parsing can significantly improve its labelled attachment score (Øvrelid and Nivre, 2007).

A more language-specific system for animacy classification has also been developed for Japanese (Baker and Brew, 2010). In this work, vari-

Table 1: Overview of the animacy tag set from Zaenen et al. (2004) with examples from the GMB.

Tag	Description	Examples
HUM	Human	Mr. **Calderon** said Mexico has become a worldwide leader ...
ORG	Organization	Mr. Calderon said **Mexico** has become a worldwide leader ...
ANI	Animal	There are only about 1,600 **pandas** still living in the wild in China.
LOC	Place	There are only about 1,600 pandas still living in the wild in **China**.
NCN	Non-concrete	There are only about 1,600 pandas still living in the **wild** in China.
CNC	Concrete	The wind blew so much **dust** around the field today.
TIM	Time	The wind blew so much dust around the field **today**.
MAC	Machine	The astronauts attached the **robot**, called Dextre, to the ...
VEH	Vehicle	Troops fired on the two civilians riding a **motorcycle** ...

ous language-specific heuristics are used to improve coverage of, e.g., loanwords from English. The features used are mainly frequency counts of nouns as subjects or objects of certain verbs. This is then fed to a Bayesian classifier, which yields quite good results on both Japanese and English.

Taking these works into account, it is clear that the use of morphosyntactic features can provide relevant information for the task of animacy classification. However, both of these approaches use binary classification schemes. It is therefore not clear whether acceptably good results could be obtained for more elaborate schemes.

3.2 Exploiting lexico-semantic resources

Orăsan & Evans (2007) present an animacy classifier which is based on knowledge obtained from WordNet (Miller, 1995). In one approach, they base this on the so-called *unique beginners* at the top of the WordNet hierarchy. The fact that some of these are closely related to animacy is then used to infer the animacy of their hyponyms. The inclusion of the classifications obtained by this system for the task of anaphora resolution is shown to improve its results.

An animacy classifier based on exploiting synonymy relations in addition to hyponymy and hyperonymy has been described for Basque (de Illaraza et al., 2002). In this work, a small set consisting of 100 nouns was manually annotated. Using an electronic dictionary from which semantic relations could be inferred, they then further automatically annotated all common nouns in a 1 million word corpus.

An approach to animacy classification for Dutch is presented in Bloem & Bouma (to appear). This approach exploits a lexical semantic resource, from which word-senses were obtained and merged per lemma. This is done, as they postulate that ambiguity in animacy per lemma ought to be relatively rare. Each lemma was then assigned a simplified animacy class depending on its animacy category – either *human*, *non-human* or *inanimate*. Similarly to Baker & Brew (2010), they also use dependency features obtained from an automatically parsed corpus for Dutch. This type-based approach obtains accuracies in the low 90% range, compared to a most frequent class baseline of about 81%.

Based on the three aforementioned works, it is clear that the use of semantic relations obtained from lexico-semantic resources such as WordNet are particularly informative for the classification of animacy.

3.3 Multi-class animacy classification

An animacy classifier which distinguishes between ten different classes of animacy has been developed by Bowman & Chopra (2012). They use a simple logistic regression classifier and quite straight-forward bag-of-words and PoS features, as well as subject, object and PP dependencies. These are obtained from the aforementioned Switchboard corpus, for which they obtain quite good results.

A quite involved system for animacy classification based on using an ensemble of voters is presented by Moore et al. (2013). This system draws its strengths from the fact that it, rather than defining and using a large number of features and training one complex classifier, uses more interpretable voting models which differ depending on the class in question. They distinguish between three categories, namely *person*, *animal* and *inanimate*. The voters comprise a variety of systems, based on the *n*-gram list method of Ji and Lin (2009), a WordNet-based approach similar to Orăsan & Evans (2007), and several others. Their results yield animacy detection rates in the mid-90% range, and can therefore be seen as an improvement upon the state of the art. However, comparison between animacy classification systems is not all that straight-forward, considering the disparity between the data sets and classification schemes used.

These two works show that multi-class animacy classification can be successfully done both with syntactic and semantic features.

4 Data

Two annotated corpora are used in this work. A further data source is concreteness ratings obtained through manual annotation (Brysbaert et al., 2013), and is used as a feature in the classifier. These ratings were obtained for approximately 40,000 English words and two-word expressions, through the use of internet crowd-sourcing. The rating was given on a five-point scale, ranging from abstract, or *language based*, to concrete, or *experience based* (Brysbaert et al., 2013).

4.1 The NXT Switchboard Corpus

Firstly, the classifier is trained and evaluated on the Switchboard corpus, as this allows for direct comparison of results to at least one previous approach (i.e. Bowman & Chopra (2012)).

4.1.1 Pre-processing of spoken data

The fact that the Switchboard corpus consists of transcribed spoken data presents challenges for some of the tools used in the feature extraction process. The primary concern identified, apart from the differing form of spoken language as compared to written language, is the presence of disfluency markers in the transcribed texts. As a preprocessing step, all disfluencies were removed using a simple automated script. Essentially, this consisted of removing all words tagged as interjections (labelled with the tag *UH*), as this is the tag assigned to disfluencies in the Switchboard corpus. Although interjections generally can be informative, the occurrences of interjections within NPs was restricted to usage as disfluencies.

4.2 The Groningen Meaning Bank

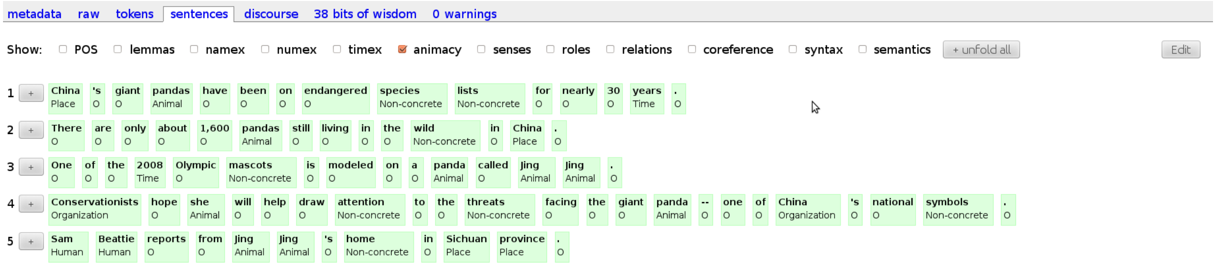
There are several corpora of reasonable size which include semantic annotation on some level, such as PropBank (Palmer et al., 2005), FrameNet (Baker et al., 1998), and the Penn Discourse TreeBank (Prasad et al., 2005). The combination of several levels of semantic annotation into one formalism are not common, however. Although some efforts exist, they tend to lack a level of formally grounded "deep" semantic representation which combines these layers.

The Groningen Meaning Bank (GMB) contains a substantial collection of English texts with such deep semantic annotation (Basile et al., 2012a). One of its goals is to combine semantic phenomena into a single formalism, as opposed to dealing with single phenomena in isolation. This provides a better handle on explaining dependencies between various ambiguous linguistic phenomena.

Manually annotating a comprehensive corpus with gold-standard semantic representations is obviously a hard and time-consuming task. Therefore, a sophisticated bootstrapping approach is used. Existing NLP tools are used to get a reasonable approximation of the target annotations to start with. Pieces of information coming from both experts (linguists) and crowd sourcing methods are then added in to improve the annotation. The addition of animacy annotation is done in the same manner. First, the animacy classifier will be incorporated into this toolchain. We then correct the tags for a subset of the corpus, which is also used to evaluate the classifier. Note that the classifier used in the toolchain uses a different model from the conditions where we evaluate on the Switchboard corpus. For the GMB, we include training data obtained through the crowd-sourcing game Wordrobe, which uses a subset of the data from the GMB (Venhuizen et al., 2013).

4.2.1 Annotation

So as to allow for evaluation of the classifier on a widely used semantically annotated corpus, one part (*p00*) of the GMB was semi-manually annotated for animacy, although this might lead to a bias with potentially overly good results for our classifier, if annotators are affected by its output. We use the tagset presented by Zaenen et al. (2004), which is given in Table 1. This tagset was chosen for the addition of animacy annotation to the GMB. Including this level of annotation



1	China	's	giant	pandas	have	been	on	endangered	species	lists	for	nearly	30	years	.
	Place	O	O	Animal	O	O	O	O	Non-concrete	Non-concrete	O	O	O	Time	O

2	There	are	only	about	1,600	pandas	still	living	in	the	wild	in	China	.
	O	O	O	O	O	Animal	O	O	O	O	Non-concrete	O	Place	O

3	One	of	the	2008	Olympic	mascots	is	modeled	on	a	panda	called	Jing	Jing	.
	O	O	O	Time	O	Non-concrete	O	O	O	O	Animal	O	Animal	Animal	O

4	Conservationists	hope	she	will	help	draw	attention	to	the	threats	facing	the	giant	panda	--	one	of	China	's	national	symbols	.
	Organization	O	Animal	O	O	O	Non-concrete	O	O	Non-concrete	O	O	O	Animal	O	O	O	Organization	O	O	Non-concrete	O

5	Sam	Beattie	reports	from	Jing	Jing	's	home	in	Sichuan	province	.
	Human	Human	O	O	Animal	Animal	O	Non-concrete	O	Place	Place	O

Figure 1: A tagged document in the GMB.

in a resource which already contains other semantic annotation should prove particularly useful, as this allows animacy to be used in conjunction with other semantically based features in NLP tools and algorithms. This annotation was done using the GMB's interface for expert annotation (Basile et al., 2012b). A total of 102 documents, containing approximately 15,000 tokens, were annotated by an expert annotator, who corrected the tags assigned by the classifier. We assign animacy tags to all nouns and pronouns. Similarly to our tagging convention for named entities, we assign the same tag to the whole NP, so that *wagon driver* is tagged with HUM, although *wagon* in isolation would be tagged with CNC. This has the added advantage that this is the manner in which NPs are annotated in the Switchboard corpus, making evaluation and comparison with Bowman & Chopra (2012) somewhat more straight-forward. An example of a tagged document can be seen in Figure 1. Table 2 shows the amount of annotated nouns per class. In order to verify the integrity of this annotation, two other experts annotated a random selection of ten documents. Inter-annotator agreement was calculated using Fleiss' kappa on this selection, yielding a score of $\kappa = .596$.

Table 2: Annotation statistics for *p00* of the GMB

HUM	NCN	CNC	TIM	ORG	LOC	ANI	VEH	MAC
1436	2077	79	500	887	512	67	28	0

5 Method

5.1 Classifiers

We experiment using four different classifiers (see Table 3). All classifiers used are obtained from the implementations provided by SciKit-learn (Pedregosa et al., 2011). For each type of classifier, we train one classifier for each class in a one-versus-all fashion. For source code, trained models and scripts to run the experiments in this paper, please see `https://github.com/bjerva/animacy`.

The classifiers are trained on a combination of the Switchboard corpus and data gathered from Wordrobe, depending on the experimental condition. In addition to the features explained below, the classifier exploits named entity tags, in that these override the proposed animacy tag where applicable. That is to say, if a named entity has already been identified and tagged as, e.g., a person, this is reflected in the animacy layer with the HUM tag.

Considering that the balance between samples per class is quite skewed, an attempt was made at placing lower weights on the samples from the majority classes. Although this did lead to a marginal increase in accuracy for the minority classes, overall accuracy dropped to such an extent that this weighting was not used for the results presented in this work.

5.2 Features

In this section, an overview of the features used by the classifiers is given.

5.2.1 Bag-of-words feature

The simplest feature used consists of looking at each lemma in the NP to be classified, and their corresponding PoS tags. We also experimented with using whole sentences as context for classification, but as this worsened results on our development data, it was not used for the evaluations later in the paper.

5.2.2 Concreteness ratings

Considering that two of the categories in our tag set discriminate between concrete and non-concrete entities, we include concreteness ratings

Table 3: Overview of the classifiers used in the experiments.

Classifier	Reference	Parameter settings
Logistic Regression (MaxEnt)	(Berger et al., 1996)	$\ell 2$ regularization
Support Vector Machine (SVM)	(Joachims, 1998)	linear kernel
Stochastic Gradient Descent (SGD)	(Tsuruoka et al., 2009)	$\ell 2$ regularization, hinge loss
Bernoulli Naive Bayes (B-NB)	(McCallum et al., 1998)	–

as a feature in the classifier (Brysbaert et al., 2013). In its original form, these ratings are quite fine-grained as they are provided with the average concreteness score given by annotators on a scale. We experimented with using different granularities of these scores as a feature. A simple binary distinction where anything with a score of $c > 2.5$ being represented as concrete, and $c \leq 2.5$ being represented as non-concrete yielded the best results, and is used in the evaluations in this paper.

5.2.3 WordNet distances

We also include a feature based on WordNet distances. In this work, we use the path distance similarity measure provided in NLTK (Bird, 2006). In essence, this measure provides a score based on the shortest path that connects the senses in a hypernym/hyponym taxonomy. First, we calculate the distance to each hypernym of every given word. These distances are then summed together for each animacy class. Taking the most frequent hypernym for each animacy class gives us the following hypernyms: *person.n.01*, *abstraction.n.06*, *city.n.01*, *time_period.n.01*, *car.n.01*, *organization.n.01*, *artifact.n.01*, *animal.n.01*, *machine.n.01*, *buddy.n.01*. The classifier then uses whichever of these words is closest as its WordNet feature.

5.2.4 Thematic roles

The use of thematic roles for animacy annotation constitutes a novel contribution from this work. Intuitively this makes sense, as e.g. agents tend to be animate. Although the GMB contains an annotation layer with thematic roles, the Switchboard corpus does not. In order to use this feature, we therefore preprocessed the latter using Boxer (Bos, 2008). We use the protoroles obtained from Boxer, namely *agent*, *theme* and *patient*. Although automatic annotation does not provide 100% accuracy, especially on such a particular data set, this feature proved somewhat useful (see Section 6.1.2).

6 Results

6.1 Evaluation on the Switchboard corpus

We employ 10-fold cross validation for the evaluations on the Switchboard corpus. All NPs were automatically extracted from the pre-processed corpus, put into random order and divided into ten equally-sized folds. In each of the ten cross validation iterations, one of these folds was left out and used for evaluation. For the sake of conciseness, averaged results over all classes are given in the comparisons of Section 6.1.1 and Section 6.1.2, whereas detailed results are only given for the best performing classifier. Note that the training data from Wordrobe is not used for the evaluations on the Switchboard corpus, as this would prohibit fair evaluation with previous work.

6.1.1 Classifier evaluation

We first ran experiments to evaluate which of the classifiers performed the best on this task. Figure 2 shows the average accuracy for each classifier, using 10-fold cross validation on the Switchboard corpus. Table 4 contains the per-class results from the cross validation performed with the best performing classifier, namely the Logistic Regression classifier. The remaining evaluations in this paper are all carried out with this classifier. Average accuracy over the 10 folds was 85.8%. This is well above the baseline of always picking the most common class (HUM), which results in an accuracy of 45.3%. More interestingly, this is somewhat higher than the best results for this dataset reported in the literature (84.9% without cross validation (Bowman and Chopra, 2012)).

6.1.2 Feature evaluation

Using the best performing classifier, we ran experiments to evaluate how different features affect the results. These experiments were also performed using 10-fold cross validation on the Switchboard corpus. Table 5 shows scores from using only one

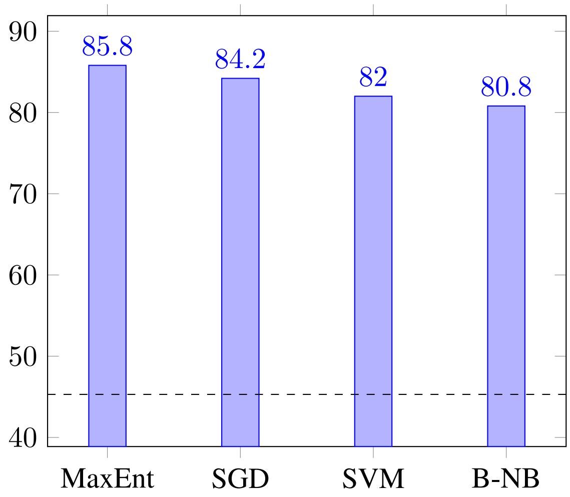


Figure 2: Accuracy of the classifiers, using 10-fold cross validation on the Switchboard corpus. The dashed line represents the most frequent class baseline.

feature in addition to the lemma and PoS of the head of the NP to be classified. Although none of the features in isolation add much to the performance of the classifier, some marginal gains can be observed.

Table 5: Comparison of the effect of including single features, from cross validation on the Switchboard corpus. All conditions consist of the feature named in the condition column in addition to Lemma+PoS.

Condition	**Precision**	**Recall**	**F-score**
Lemma+PoS	0.846	0.850	0.848
Bag of Words	0.851	0.856	0.853
Concreteness	0.847	0.851	0.849
WordNet	0.849	0.855	0.852
Thematic Roles	0.847	0.851	0.849
All features	0.851	0.857	0.854

6.1.3 Performance on unknown words

For a task such as animacy classification, where many words can be reliably classified based solely on their lemma and PoS tag, it is particularly interesting to investigate performance on unknown words. As in all other conditions, this was evaluated using 10-fold cross validation on the Switchboard corpus. It should come as no surprise that the results are substantially below those for known words, for every single class. The average accuracy for this condition was 59.2%, which can be compared to the most frequent class (NCN) baseline at 43.0%.

6.2 Evaluation on the GMB

Since one of the purposes of the development of this classifier was to include it in the tools used in the tagging of the GMB, we also present the first results in the literature for the animacy annotation of this corpus. Due to the limited size of the portion of this corpus for which animacy tags have been manually corrected, no cross-validation was performed. However, due to the high differences in the training data from the Switchboard corpus, and the evaluation data in the GMB, the results could be seen as a lower bound for this classifier on this data set. Table 4 contains the results from this evaluation. The accuracy on this dataset was 79.4%, which can be compared to a most frequent class baseline of 37.2%.

6.3 Excluding pronouns

The discrepancy between the results obtained from the Switchboard corpus and the GMB does call for some investigation. Considering that the Switchboard corpus consists of spoken language, it contains a relatively large amount of personal pronouns compared to, e.g., news text. Taking into account that these pronouns are rarely ambiguous as far as animacy is concerned, it seems feasible that this may be why the results for the Switchboard corpus are better than those of the GMB. To evaluate this, a separate experiment was run in which all pronouns were excluded. As a large amount of pronouns are tagged as HUM, the F-scores for this class dropped by 8% and 5% for the Switchboard corpus and GMB respectively. For the GMB, results for other classes remained fairly stable, most likely due to there not being many pronouns present which affect the remaining classes. For the Switchboard corpus, however, an increase in F-score was observed for several classes. This might be explained by that the exclusion of pronouns lowered the classifier's pre-existing bias for the HUM class, as the number of annotated examples was lowered from approximately 85,000 to 15,000.

Animacy classification of pronouns can be considered trivial, as there is little or no ambiguity of that the referent of e.g. *he* is HUM. Even so, pronouns were included in the main results provided

Table 4: Results from 10-fold cross validation on the Switchboard corpus and evaluation on the GMB.

	Switchboard				GMB			
Class	**Count**	**Precision**	**Recall**	**F-score**	**Count**	**Precision**	**Recall**	**F-score**
HUM	82596	0.91	0.97	0.94	1436	0.82	0.79	0.80
NCN	62740	0.82	0.94	0.88	2077	0.76	0.88	0.82
CNC	12425	0.75	0.43	0.55	79	0.48	0.13	0.20
TIM	7179	0.88	0.85	0.87	500	0.77	0.95	0.85
ORG	6847	0.71	0.26	0.38	887	0.85	0.68	0.75
LOC	5592	0.71	0.66	0.69	512	0.89	0.71	0.79
ANI	2362	0.89	0.36	0.51	67	0.63	0.22	0.33
VEH	1840	0.89	0.45	0.59	28	1.00	0.39	0.56
MAC	694	0.80	0.34	0.47	-	-	-	-
MIX	34	0.00	0.00	0.00	-	-	-	-

here, as this is the standard manner of reporting results in prior work.

6.4 Summary of results

Table 6 contains a brief overview of the most essential results from this work. For the Switchboard corpus, this constitutes the current best results in the literature. As for the GMB, this constitutes the first results in the literature for animacy classification.

Table 6: Main results from all conditions. B&C (2012) refers to Bowman & Chopra (2012).

Corpus	**Condition**	**Accuracy**
Switchboard	B&C (2012)	0.849
	Unknown words	0.592
	Known words	0.860
	All words	0.858
GMB	Unknown words	0.764
	Known words	0.831
	All words	0.794

7 Discussion

The work presented in this paper constitutes a minor improvement to the previously best results for multi-class animacy classification on the Switchboard corpus (Bowman and Chopra, 2012). Additionally, we also present the first results in the literature for animacy classification on the GMB, allowing for future research to use this work as a point of comparison. It is, however, important to note that the results obtained for the GMB in this paper are prone to bias, as the annotation procedure was done in a semi-automatic fashion. If annotators were affected by the output of the classifier, this is likely to have improved the results presented here.

A striking factor when observing the results, is the high discrepancy in performance between the GMB and the Switchboard corpus. This is, however, not all that surprising. Considering that the Switchboard corpus consists of spoken language, and the GMB contains written language, one can easily draw the conclusion that the domain differences pose a substantial obstacle. This can, for instance, be seen in the differing vocabulary. In the cross-validation conditions for the Switchboard corpus, approximately 1% of the words to be classified in each fold are unknown to the classifier. As for the GMB, approximately 10% of the words are unknown. As mentioned in Section 6.1.2, the lemma of the head noun in an NP is a very strong feature, which naturally can not be used in the case of unknown words. As seen in Table 6, performance on known words in the GMB is not far away from that of known words in the Switchboard corpus.

Although a fairly good selection of classifiers were tested in this work, there is room for improvement in this area. The fact that the Logistic Regression classifier outperformed all other classifiers is likely to have been caused by that not enough effort was put into parameter selection for the other classifiers. More sophisticated classifiers, such as Artificial Neural Networks, ought to at the very least replicate the results achieved here. Quite likely, results should even improve, seeing that the added computational power of ANNs allows us to capture more interesting/deeper statistical patterns, if they exist in the data.

The features used in this paper mainly revolved around semantically oriented ones, such as semantic relations from WordNet, thematic roles and, arguably, concreteness ratings. Better results could most likely be achieved if one also incorporated more syntactically oriented features, such as frequency counts from a dependency parsed corpus, as done by e.g. Bowman & Chopra (2012) and Øvrelid (2009). Other options include the use of more linguistically motivated features, such as exploiting relative pronouns (i.e. *who* vs. *which*).

8 Conclusions and future work

At the beginning of this paper, we set out three aims. Firstly, we wanted to improve upon the state of the art in multi-class animacy classification. A conclusive statement to that effect is hard to make, considering that comparison was only made directly to one previous work. However, as our performance compared to this work was somewhat higher, this work certainly marks some sort of improvement. Secondly, we aimed at investigating whether a corpus of spoken language containing animacy annotation could be used to annotate a corpus of written language. As our results for the GMB are well above the baseline, we conclude that this is indeed feasible, in spite of the disparities between language form and vocabulary. Lastly, we aimed at using the resulting classifier as a part of the toolchain used to annotate the GMB. This goal has also been met.

As for future work, the fact that animacy is marked explicitly in many languages presents a golden opportunity to alleviate the annotation of this semantic property for languages in which it is not explicitly marked. By identifying these markers, the annotation of animacy in such a language should be relatively trivial through the use of parallel texts. Alternatively, one could look at using existing annotated corpora, such as Talbanken05 (Nivre et al., 2006), as a source of annotation. One could then look at transferring this annotation to a second language. Although intuitively promising, this approach has some potential issues, as animacy is not represented universally across languages. For instance, fluid containers (e.g. cups, spoons) represent a class of nouns which are considered grammatically animate in Algonquian (Quinn, 2001). Annotating such items as animate in English would most likely not be considered correct, neither by native speakers nor by most experts. Nevertheless, if a sufficiently large amount of languages have some manner of consensus as to where a given entity is in an animacy hierarchy, this problem ought to be solvable by simply hand-picking such languages.

References

Kirk Baker and Chris Brew. 2010. Multilingual animacy classification by sparse logistic regression. *OSUWPL*, 59:52–75.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics. Proceedings of the Conference*, pages 86–90, Université de Montréal, Montreal, Quebec, Canada.

Valerio Basile, Johan Bos, Kilian Evang, and Noortje Venhuizen. 2012a. Developing a large semantically annotated corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, pages 3196–3200, Istanbul, Turkey.

Valerio Basile, Johan Bos, Kilian Evang, and Noortje Venhuizen. 2012b. A platform for collaborative semantic annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 92–96, Avignon, France.

Adam L Berger, Vincent J Della Pietra, and Stephen A Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71.

Steven Bird. 2006. NLTK: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Linguistics.

Jelke Bloem and Gosse Bouma. to appear. Automatic animacy classification for dutch. *Computational Linguistics in the Netherlands Journal*, 3, 12/2013.

Johan Bos. 2008. Wide-Coverage Semantic Analysis with Boxer. In J. Bos and R. Delmonte, editors, *Semantics in Text Processing. STEP 2008 Conference Proceedings*, volume 1 of *Research in Computational Semantics*, pages 277–286. College Publications.

Samuel R Bowman and Harshit Chopra. 2012. Automatic animacy classification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop*, pages 7–10. Association for Computational Linguistics.

Joan Bresnan, Anna Cueni, Tatiana Nikitina, R Harald Baayen, et al. 2007. Predicting the dative alternation. *Cognitive foundations of interpretation*, pages 69–94.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2013. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior Research Methods*, pages 1–8.

Sasha Calhoun, Jean Carletta, Jason M Brenier, Neil Mayo, Dan Jurafsky, Mark Steedman, and David Beaver. 2010. The nxt-format switchboard corpus: a rich resource for investigating the syntax, semantics, pragmatics and prosody of dialogue. *Language Resources and Evaluation*, 44(4):387–419.

Bernard Comrie. 1989. *Language universals and linguistic typology: Syntax and morphology*. University of Chicago press.

Östen Dahl and Kari Fraurud. 1996. Animacy in grammar and discourse. *PRAGMATICS AND BEYOND NEW SERIES*, pages 47–64.

Arantza Díaz de Illaraza, Aingeru Mayor, and Kepa Sarasola. 2002. Semiautomatic labelling of semantic features. In *Proceedings of the 19th International Conference on Computational Linguistics*.

Felice Dell'Orletta, Alessandro Lenci, Simonetta Montemagni, and Vito Pirrelli. 2005. Climbing the path to grammar: A maximum entropy model of subject/object learning. In *Proceedings of the Workshop on Psychocomputational Models of Human Language Acquisition*, pages 72–81. Association for Computational Linguistics.

Heng Ji and Dekang Lin. 2009. Gender and animacy knowledge discovery from web-scale n-grams for unsupervised person mention detection. In *PACLIC*, pages 220–229.

Thorsten Joachims. 1998. *Text categorization with support vector machines: Learning with many relevant features*. Springer.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules.

Andrew McCallum, Kamal Nigam, et al. 1998. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Joshua L. Moore, Christopher J.C. Burges, Erin Renshaw, and Yih Wen-tau. 2013. Animacy detection with voting models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 55–60. Association for Computational Linguistics.

Joakim Nivre, Jens Nilsson, and Johan Hall. 2006. Talbanken05: A swedish treebank with phrase structure and dependency annotation. In *Proceedings of the fifth International Conference on Language Resources and Evaluation (LREC)*, pages 1392–1395.

Constantin Orăsan and Richard Evans. 2007. Np animacy identification for anaphora resolution. *J. Artif. Intell. Res.(JAIR)*, 29:79–103.

Lilja Øvrelid and Joakim Nivre. 2007. When word order and part-of-speech tags are not enough–Swedish dependency parsing with rich linguistic features. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, pages 447–451.

Lilja Øvrelid. 2005. Animacy classification based on morphosyntactic corpus frequencies: some experiments with norwegian nouns. In *Proc. of the Workshop on Exploring Syntactically Annotated Corpora*.

Lilja Øvrelid. 2006. Towards robust animacy classification using morphosyntactic distributional features. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 47–54. Association for Computational Linguistics.

Lilja Øvrelid. 2009. Empirical evaluations of animacy annotation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 630–638. Association for Computational Linguistics.

Martha Palmer, Paul Kingsbury, and Daniel Gildea. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Rashmi Prasad, Aravind Joshi, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, and Bonnie Webber. 2005. The Penn Discourse TreeBank as a resource for natural language generation. In *Proc. of the Corpus Linguistics Workshop on Using Corpora for Natural Language Generation*, pages 25–32.

Conor Quinn. 2001. A preliminary survey of animacy categories in penobscot. In *Papers of the 32nd. Algonquian Conference*, pages 395–426.

Anette Rosenbach. 2008. Animacy and grammatical variation–findings from English genitive variation. *Lingua*, 118(2):151–171.

Anatol Stefanowitsch. 2003. Constructional semantics as a limit to grammatical alternation: The two genitives of English. *TOPICS IN ENGLISH LINGUISTICS*, 43:413–444.

Yoshimasa Tsuruoka, Jun'ichi Tsujii, and Sophia Ananiadou. 2009. Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, volume 1, pages 477–485. Association for Computational Linguistics.

Noortje J. Venhuizen, Valerio Basile, Kilian Evang, and Johan Bos. 2013. Gamification for word sense labeling. *Proc. 10th International Conference on Computational Semantics (IWCS-2013)*, pages 397–403.

Annie Zaenen, Jean Carletta, Gregory Garretson, Joan Bresnan, Andrew Koontz-Garboden, Tatiana Nikitina, M Catherine O'Connor, and Tom Wasow. 2004. Animacy encoding in english: why and how. In *Proceedings of the 2004 ACL Workshop on Discourse Annotation*, pages 118–125. Association for Computational Linguistics.

Using Minimal Recursion Semantics for Entailment Recognition

Elisabeth Lien
Department of Informatics, University of Oslo
elien@ifi.uio.no

Abstract

This paper describes work on using Minimal Recursion Semantics (MRS) representations for the task of recognising textual entailment. I use entailment data from a SemEval-2010 shared task to develop and evaluate an entailment recognition heuristic. I compare my results to the shared task winner, and discuss differences in approaches. Finally, I run my system with multiple MRS representations per sentence, and show that this improves the recognition results for positive entailment sentence pairs.

1 Introduction

Since the first shared task on Recognising Textual Entailment (RTE) (Dagan et al., 2005) was organised in 2005, much research has been done on how one can detect entailment between natural language sentences. A range of methods within statistical, rule based, and logical approaches have been applied. The methods have exploited knowledge on lexical relations, syntactic and semantic knowledge, and logical representations.

In this paper, I examine the benefits and possible disadvantages of using rich semantic representations as the basis for entailment recognition. More specifically, I use Minimal Recursion Semantics (MRS) (Copestake et al., 2005) representations as output by the English Resource Grammar (ERG) (Flickinger, 2000). I want to investigate how logical-form semantics compares to syntactic analysis on the task of determining the entailment relationship between two sentences. To my knowledge, MRS representations have so far not been extensively used for this task.

To this end, I revisit a SemEval shared task from 2010 that used entailment recognition as a means to evaluate parser output. The shared task data were constructed so as to require only syntactic analysis to decide entailment for a sentence pair. The MRSs should perform well on such data, as they abstract over irrelevant syntactic variation, as for example use of active vs. passive voice, or meaning-preserving variation in constituent order, and thus normalise at a highly suitable level of "who did what to whom". The core idea of my approach is graph alignment over MRS representations, where successful alignment of MRS nodes is treated as an indicator of entailment.

This work is part of an ongoing dissertation project, where the larger goal is to look more closely at correspondences between logical and textual entailment, and the use of semantic representations in entailment recognition.

Besides using MRS, one novel aspect of this work is an investigation of using n-best lists of parser outputs in deciding on entailment relations. In principle, the top-ranked (i.e., most probable) parser output should correspond to the intended reading, but in practise this may not always be the case. To increase robustness in our approach to imperfect parse ranking, I generalise the system to operate over n-best lists of MRSs. This setup yields greatly improved system performance and advances the state of the art on this task, i.e., makes my system retroactively the top performer in this specific competition.

The rest of this paper is organised as follows: in section 2, I describe the task of recognising textual entailment. I also briefly describe MRS representations, and mention previous work on RTE using MRS. In section 3, I analyse the shared task data, and implement an entailment decision component which takes as input MRS representations from the ERG. I then analyse the errors that the component makes. Finally, I compare my results to the actual winner of the 2010 shared task. In section 4, I generalise my approach to 10-best lists of MRSs.

Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 76–84,
Gothenburg, Sweden, April 26-30 2014. ©2014 Association for Computational Linguistics

2 Background

In the following, I briefly review the task of recognising entailment between natural language sentences. I also show an example of an MRS representation, and mention some previous work on entailment recognition that has used MRSs.

2.1 Recognising Textual Entailment

Research on automated reasoning has always been a central topic in computer science, with much focus on logical approaches. Although there had been research on reasoning expressed in natural language, the PASCAL Recognising Textual Entailment (RTE) Challenge (Dagan et al., 2005) spurred wide interest in the problem. In the task proposed by the RTE Challenge, a system is required to recognise whether the meaning of one text can be inferred from the meaning of another text. Their definition of inference, or *textual entailment*, is based on the everyday reasoning abilities of humans rather than the logical properties of language.

The RTE Challenge evolved from the relatively simple task of making binary decisions about sentence pairs into more complex variants with many categories and multi-sentence texts. The data sets issued by the organisers over the years provide valuable research material. However, they contain a wide range of inference phenomena, and require both ontological and world knowledge. The data set that I have used for the present work, the PETE data set, focusses on syntactic phenomena, and does not require any knowledge about the state of the world or ontological relations.

2.2 Minimal Recursion Semantics

Minimal Recursion Semantics (MRS) (Copestake et al., 2005) is a framework for computational semantics which can be used for both parsing and generation. MRS representations are expressive, have a clear interface with syntax, and are suitable for processing. MRSs can be underspecified with regard to scope in order to allow a semantically ambiguous sentence to be represented with a single MRS that captures every reading. MRS is integrated with the HPSG English Resource Grammar (ERG) (Flickinger, 2000).

An MRS representation contains a multiset of relations, called *elementary predications* (EPs). An EP usually corresponds to a single lexeme, but can also represent general grammatical features. Each EP has a *predicate symbol* which, in the case of lexical predicates, encodes information about lemma, part-of-speech, and sense distinctions. An EP also has a *label* (also called *handle*) attached to it. Each EP contains a list of numbered arguments: `ARG0`, `ARG1`, etc. The value of an argument can be either a scopal variable (a handle which refers to another EP's label) or a non-scopal variable (events or states, or entities).

The `ARG0` position of the argument list has the EP's *distinguished variable* as its value. This variable denotes either an event or state, or a referential or abstract entity (e_i or x_i, respectively). Each non-quantifier EP has its unique distinguished variable.

Finally, an MRS has a set of *handle constraints* which describe how the scopal arguments of the EPs can be equated with EP labels. A constraint $h_i =_q h_j$ denotes equality modulo quantifier insertion. In addition to the indirect linking through handle constraints, EPs are directly linked by sharing the same variable as argument values. The resulting MRS forms a connected graph.

In figure 2, we see an MRS for the sentence *Somebody denies there are barriers* from the PETE development data (id 4116)[1]. The topmost relation of the MRS is `_deny_v_to`, which has two non-empty arguments: x_5 and h_{10}. x_5 is the distinguished variable of the relations `_some_q` and `person`, which represent the pronoun *somebody*. A handle constraint equates the sentential variable h_{10} with h_{11}, which is the label of `_be_v_there`. This last relation has x_{13} as its sole argument, which is the distinguished variable of `udef_q` and `_barrier_n_to`, the representation of *barriers*.

2.3 Previous Work on RTE using MRS

To my knowledge, MRS has not been used extensively in entailment decision systems. Notable examples of approaches that use MRSs are Wotzlaw and Coote (2013), and Bergmair (2010).

In Wotzlaw and Coote (2013), the authors present an entailment recognition system which combines high-coverage syntactic and semantic text analysis with logical inference supported by relevant background knowledge. Their system combines deep and shallow linguistic analysis, and transforms the results into scope-resolved

[1] The event and entity variables of the EPs often have grammatical features attached to them. I have removed these features from the MRS for the sake of readability.

$$\begin{array}{l}
\langle\, h_1, \\
\quad h_4\text{:proper\_q}\langle 0{:}5\rangle\text{(ARG0 } x_6\text{, RSTR } h_5\text{, BODY } h_7\text{),} \\
\quad h_8\text{:named}\langle 0{:}5\rangle\text{(ARG0 } x_6\text{, CARG } \textit{Japan}\text{),} \\
\quad h_2\text{:\_deny\_v\_to}\langle 6{:}12\rangle\text{(ARG0 } e_3\text{, ARG1 } x_6\text{, ARG2 } h_{10}\text{, ARG3 } i_9\text{),} \\
\quad h_{11}\text{:\_be\_v\_there}\langle 19{:}22\rangle\text{(ARG0 } e_{12}\text{, ARG1 } x_{13}\text{),} \\
\quad h_{14}\text{:udef\_q}\langle 23{:}37\rangle\text{(ARG0 } x_{13}\text{, RSTR } h_{15}\text{, BODY } h_{16}\text{),} \\
\quad h_{17}\text{:\_real\_a\_1}\langle 23{:}27\rangle\text{(ARG0 } e_{18}\text{, ARG1 } x_{13}\text{),} \\
\quad h_{17}\text{:\_barrier\_n\_to}\langle 28{:}37\rangle\text{(ARG0 } x_{13}\text{, ARG1 } i_{19}\text{)} \\
\{\, h_{15} =_q h_{17},\, h_{10} =_q h_{11},\, h_5 =_q h_8,\, h_1 =_q h_2 \,\}\,\rangle
\end{array}$$

Figure 1: MRS for the sentence *Japan denies there are real barriers.*

$$\begin{array}{l}
\langle\, h_1, \\
\quad h_4\text{:person}\langle 0{:}8\rangle\text{(ARG0 } x_5\text{),} \\
\quad h_6\text{:\_some\_q}\langle 0{:}8\rangle\text{(ARG0 } x_5\text{, RSTR } h_7\text{, BODY } h_8\text{),} \\
\quad h_2\text{:\_deny\_v\_to}\langle 9{:}15\rangle\text{(ARG0 } e_3\text{, ARG1 } x_5\text{, ARG2 } h_{10}\text{, ARG3 } i_9\text{),} \\
\quad h_{11}\text{:\_be\_v\_there}\langle 22{:}25\rangle\text{(ARG0 } e_{12}\text{, ARG1 } x_{13}\text{),} \\
\quad h_{14}\text{:udef\_q}\langle 26{:}35\rangle\text{(ARG0 } x_{13}\text{, RSTR } h_{15}\text{, BODY } h_{16}\text{),} \\
\quad h_{17}\text{:\_barrier\_n\_to}\langle 26{:}35\rangle\text{(ARG0 } x_{13}\text{, ARG1 } i_{18}\text{)} \\
\{\, h_{15} =_q h_{17},\, h_{10} =_q h_{11},\, h_7 =_q h_4,\, h_1 =_q h_2 \,\}\,\rangle
\end{array}$$

Figure 2: MRS for the sentence *Somebody denies there are barriers.*

MRS representations. The MRSs are in turn translated into another semantic representation format, which, enriched with background knowledge, forms the basis for logical inference.

In Bergmair (2010), we find a theory-driven approach to textual entailment that uses MRS as an intermediate format in constructing meaning representations. The approach is based on the assumptions that the syllogism is a good approximation of natural language reasoning, and that a many-valued logic provides a better model of natural language semantics than bivalent logics do. MRSs are used as a step in the translation of natural language sentences into logical formulae that are suitable for processing. Input sentences are parsed with the ERG, and the resulting MRSs are translated into ProtoForms, which are fully recursive meaning representations that are closely related to MRSs. These ProtoForms are then decomposed into syllogistic premises that can be processed by an inference engine.

3 Recognising Syntactic Entailment using MRSs

In this section, I briefly review the SemEval-2010 shared task that used entailment decision as a means of evaluating parsers. I then describe the entailment system I developed for the shared task data, and compare its results to the winner of the original task.

3.1 The PETE Shared Task

Parser Evaluation using Textual Entailments (PETE) was a shared task in the SemEval-2010 Evaluation Exercises on Semantic Evaluation (Yuret et al., 2010). The task involved building an entailment system that could decide entailment for sentence pairs based on the output of a parser. The organisers proposed the task as an alternative way of evaluating parsers. The parser evaluation method that currently dominates the field, PARSEVAL (Black et al., 1991), compares the phrase-structure bracketing of a parser's output with the gold annotation of a treebank. This makes the evaluation both formalism-dependent and vulnerable to inconsistencies in human annotations.

The PETE shared task proposes a different evaluation method. Instead of comparing parser output directly to a gold standard, one can evaluate *indirectly* by examining how well the parser output supports the task of entailment recognition. This strategy has several advantages: the evaluation is formalism-independent, it is easier for annotators to agree on entailment than on syntactic categories and bracketing, and the task targets semantically relevant phenomena in the parser output. The data are constructed so that syntactic analysis of the sentences is sufficient to determine the entailment relationship. No background knowledge or reasoning ability is required to solve the task.

It is important to note that in the context of the PETE shared task, entailment decision is not a goal in itself, it is just a tool for parser evaluation.

The PETE organisers created two data sets for the task: a development set of 66 sentence pairs, and a test set of 301 pairs. The data sets were built by taking a selection of sentences that contain syntactic dependencies that are challenging for state-of-the-art parsers, and constructing short entailments that (in the case of positive entailment pairs) reflect these dependencies. The resulting sentence pairs were annotated with entailment judgements by untrained annotators, and only sentence pairs with a high degree of inter-annotator agreement were kept.

20 systems from 7 teams participated in the PETE task. The best scoring system was the Cambridge system (Rimell and Clark, 2010), with an accuracy of 72.4 %.

3.2 The System

My system consists of an entailment decision component that processes MRS representations as output by the ERG[2]. The entailment decision component is a Python implementation I developed after analysing the PETE development data.

The core idea is based on graph alignment, seeking to establish equivalence relations between components of MRS graphs. In a nutshell, if all nodes of the MRS corresponding to the hypothesis can be aligned with nodes of the MRS of the text, then we will call this relation MRS inclusion, and treat it as an indicator for entailment.[3] Furthermore, the PETE data set employs a limited range of "robust" generalisations in hypothesis strings, for example replacing complex noun phrases from the text by an underspecified pronoun like *somebody*. To accomodate such variation, my graph alignment procedure supports a number of "robust" equivalences, for example allowing an arbitrarily complex sub-graph to align with the graph fragment corresponding to expressions like *somebody*. These heuristic generalisations were designed in response to an in-depth analysis of the PETE development corpus, where I made the following observations for the sentences of positive entailment pairs (I use T_{sent} to mean the text sentence, and H_{sent} to mean the hypothesis sentence):

- H_{sent} is always shorter than T_{sent}.
- In some cases, H_{sent} is completely included in T_{sent}.
- Mostly, H_{sent} is a substructure of T_{sent} with minor changes:
  - T_{sent} is an active sentence, while H_{sent} is passive.
  - A noun phrase in T_{sent} has been replaced by *somebody*, *someone* or *something* in H_{sent}.
  - The whole of H_{sent} corresponds to a complex noun phrase in T_{sent}.

In addition, I noted that the determiner or definiteness of a noun phrase often changes from text to hypothesis without making any difference for the entailment. I also noted that, in accordance with the PETE design principles, the context provided by the text sentence does not influence the entailment relationship.

In the negative entailment pairs the hypothesis is usually a combination of elements from the text that does not match semantically with the text.

I examined treebanked MRS representations of the PETE development data in order to develop an entailment recognition heuristic. I found that by taking the EPs that have an *event variable* as their distinguished variable, I would capture the semantically most important relations in the sentence (the verbs). The heuristic picks out all EPs whose `ARG0` is an event variable from both the text and hypothesis MRSs—let us call them *event relations*. Then it tries to *match* all the event relations of the hypothesis to event relations in the text. In the following, T_{mrs} means the MRS for the text sentence, and H_{mrs} the MRS for the hypothesis. We say that two event relations match if:

1. they are the same or similar relations. Two event relations are the same or similar if they share the same predicate symbol, or if their predicate symbols contain the same lemma and part-of-speech.
2. and all their arguments match. Two arguments in the same argument position match if:

[2] I used the 1212 release of the ERG, in combination with the PET parser (Callmeier, 2000).

[3] On this view, bidirectional inclusion indicates that the two MRS graphs are isomorphic, i.e., logically equivalent.

- they are the same relation; or
- the argument in T_{mrs} represents a noun phrase and the argument in H_{mrs} is *somebody/someone/something*; or
- the argument in T_{mrs} is either a scopal relation or a conjunction relation, and the argument in the hypothesis is an argument of this relation; or
- the argument in H_{mrs} is not expressed.

Let us see how the heuristic works for the following sentence pair (PETE id 4116):

4116_T_{sent}: The U.S. wants the removal of what it perceives as barriers to investment; Japan denies there are real barriers.

4116_H_{sent}: Somebody denies there are barriers.

Figure 2 shows the MRS for 4116_H_{sent}. Figure 1 shows an MRS for the part of 4116_T_{sent} that entails 4116_H_{sent}: *Japan denies there are real barriers*. The heuristic picks out two relations in 4116_H_{mrs} that have an event variable as their distinguished variable: `_deny_v_to` and `_be_v_there`. It then tries to find a match for these relations in the set of event relations in 4116_T_{mrs}:

- The relation `_deny_v_to` also appears in 4116_T_{mrs}, and all its argument variables can be unified since their relations match according to the heuristic:
  - x_5 unifies with x_6, since `_some_q` and `person` (which represent *somebody*) match `proper_q` and `named` (which represent *Japan*[4])
  - h_{10} unifies with h_{10}, since they both (via the handle constraints) lead to the relation `_be_v_there`.
  - The variables i_9 and i_9 both represent unexpressed arguments, and so are trivially unified.
- The relation `_be_v_there` matches the corresponding relation in 4116_T_{mrs}, since their single argument x_{13} denotes the same relations: `udef_q` and `_barrier_n_to`.

This strategy enables us to capture all the core relations of the hypothesis. When examining the data one can see that, contrary to the design principles for the PETE data, some sentence pairs do require reasoning. The heuristic will fail to capture such pairs.

The ERG is a precision grammar and does not output analyses for sentences that are ungrammatical. Some of the sentences in the PETE data sets are arguably in a grammatical gray zone, and consequently the ERG will not give us MRS representations for such sentences. In some cases, errors in an MRS can also cause the MRS processing in the system to fail. Therefore, my system must have a fallback strategy for sentence pairs were MRSs are lacking or processing fails. The system answer NO in such cases, since it has no evidence for an entailment relationship.

For the development process I used both treebanked and 1-best MRSs.

3.3 Error analysis

Tables 1 and 2 show the entailment decision results for 1-best MRSs for the PETE development and test data. The ERG parsed 61 of the 66 pairs in the development set, and 285 of the 301 pairs in the test set. The five development set pairs that did not get a parse were all negative entailments pairs. Of the 16 test pairs that failed to parse, 10 were negative entailment pairs. The system's fallback strategy labels these as NO.

	gold YES: 38	gold NO: 28
sys YES	25	2
sys NO	13	26

Table 1: The results for 1-best MRSs for the PETE development data.

	gold YES: 156	gold NO: 145
sys YES	78	10
sys NO	78	135

Table 2: The results for 1-best MRSs for the PETE test data.

The implementation of the heuristic is fine-grained in its treatment of the transformations from text to hypothesis that I found in the PETE development sentences. Although I tried to anticipate possible variations in the test data set, it inevitably contained cases that were not covered by

[4] According to the heuristic, any proper name matches the pronoun *somebody*, so we do not have to consider the actual proper name involved.

the code. This meant that occasionally the system was not able to recognise an entailment.

However, most of the incorrect judgements were caused either by errors in the MRSs, or by features of the MRSs or the PETE sentence pairs that are outside the scope of my heuristic:

1. Recognising the entailment depends on information about coreferring expressions, which is not part of the MRS analyses.

2. The entailment (or non-entailment) relationship depends on something other than syntactic structure. Recognising the entailment requires background knowledge and reasoning. This means the entailment is really outside the stated scope of the PETE task.

3. For some of the PETE sentence pairs, the gold annotation can be discussed. The following pair (PETE id 2079) is labeled NO, but is structurally similar to sentence pairs in the data set that are labeled YES: *Also, traders are in better shape today than in 1987 to survive selling binges. ⇒ Binges are survived.*

3.4 Results and Comparison to Shared Task Winner

At this point, we are ready to compare the results with the winner of the PETE shared task. Of the 20 systems that took part in the shared task, the best scoring participant was the Cambridge system, developed by Laura Rimell and Stephen Clark of the University of Cambridge (Rimell and Clark, 2010). Their system had an overall accuracy of 72.4 %. My focus here is on comparing the performance of the entailment systems, not the parsers.

The Cambridge system: The system consists of a parser and an entailment system. Rimell and Clark used the C&C parser, which can produce output in the form of grammatical relations, that is, labelled head-dependencies. They used the parser with the Stanford Dependency scheme (de Marneffe et al., 2006), which defines a hierarchy of 48 grammatical relations.

The Cambridge entailment system was based on the assumption that the hypothesis is a simplified version of the text. In order to decide entailment, one can then compare the grammatical relations—the SDs—of the two sentences[5]. If the SDs of the hypothesis are a subset of the SDs of the text, then the text entails the hypothesis. However, because the hypotheses in the PETE data are often not a direct substructure of the text, Rimell and Clark used heuristics to deal with alterations between sentences (in the following, I use T_{sd} and H_{sd} to mean the grammatical relations of text and hypothesis sentences, respectively):

1. If a SD in the hypothesis contains a token which is not in the text, this SD is ignored. This means that passive auxiliaries, pronouns, determiners and expletive subjects that are in H_{sd} but not in T_{sd} are ignored.

2. Passive subjects are equated with direct objects. This rule handles the PETE pairs where the active verb of the text has become a passive in the hypothesis.

3. When checking whether the SDs in H_{sd} are a subset of the SDs in T_{sd}, only subject and object relations are considered (core relations).

4. The intersection of SDs in T_{sd} and H_{sd} has to be non-empty (this is not restricted to subjects and objects).

To sum up: if core(H_{sd}) $\subseteq$ core(T_{sd}) and $H_{sd} \cap T_{sd} \neq \emptyset$, then T_{sent} entails H_{sent}.

Results for 1-best (automatically generated) test data: We can now compare the results from the system for 1-best test data with those of Cambridge.

In order to compare the test data results from my system with those of Rimell & Clark, I have to account for those sentence pairs that the ERG could not parse (16) and the MRS pairs that my system could not process (1). I use the same fallback strategy as Rimell & Clark, and let the entailment decision be NO for those sentence pairs the system cannot handle. For comparison, I also include the results for SCHWA (University of Sydney), the second highest scorer of the systems that participated in the shared task.

From the results in table 3 we can see that my system would have done well in the shared task. An accuracy of 70.7 % places the system a little

[5] In Rimell and Clark (2010), the authors used the abbreviation GR to mean the grammatical relations of the Stanford Dependency scheme. I use SD instead, to avoid confusion with the term GR as used by Carroll et al. (1999)

System	A	P	R	F1
Cambridge	72.4	79.6	62.8	70.2
My system	70.7	88.6	50.0	63.9
SCHWA	70.4	68.3	80.1	73.7

Table 3: The two top systems from the PETE shared task (Yuret et al., 2010) compared to my system. Accuracy (A) gives the percentage of correct answers for both YES and NO. Precision (P), recall (R) and F1 are calculated for YES.

ahead of SCHWA, the second best system. We also note that my system has a significantly higher precision on the YES judgements than the other two systems.

Resuls for gold/treebanked development data: In order to evaluate their entailment system, Rimell & Clark ran their system on manually annotated grammatical relations. Given a valid entailment decision approach and correct SDs, the system could in theory achieve 100 % accuracy. Cambridge achieved 90.9 % accuracy on these gold data. The authors noted that one incorrect decision was due to a PETE pair requiring coreference resolution, three errors were caused by certain transformations between text and hypothesis that were not covered by their heuristic, and two errors occured because the heuristic ignored some SDs that were crucial for recognising the entailments.

When I ran my system on treebanked MRSs for the PETE development data, it achieved an accuracy of 92.4 %, which is slightly better than the accuracy for Cambridge.

MRSs vs. grammatical relations: The information that the Cambridge system uses is word dependencies that are typed with grammatical relations. More specifically, Cambridge uses subject and object relations between words to decide entailment. Because the relations are explicit—we know exactly what type of grammatical relation that holds between two words—it is easy to select the relations in H_{sd} that one wants to check.

The EPs of MRSs are a mixture of lexical relations, and various syntactic and semantic relations. A lot of the grammatical information that is explicitly represented as SDs in the Stanford scheme is implicitly represented in MRS EPs as argument-value pairs. For example, the subject relation between *he* and the verb in *he runs* is represented as `(nsubj run he)` in Stanford notation. The corresponding representation in an MRS is `[ run_v_1 LBL: h ARG0: e ARG1: x ]`, where `ARG1` denotes the proto-agent of the verb. The assignment of semantic roles to arguments in EPs is not affected by passivisation or dative shift, whereas such transformations can cause differences in SDs. For sentence pairs where these phenomena occur, it is easier to match EPs and their arguments than the corresponding grammatical relations.

Cambridge heuristic vs. my heuristic: The Cambridge system checks whether the subject and object relations in H_{sd} also appear in T_{sd}. However, because their heuristic ignores tokens in the hypothesis that are not in the text, the system in certain cases does not check core relations that are crucial to the entailment relationship.

My system checks whether the event relations in H_{mrs} also appear in T_{mrs}, and whether their arguments can be matched. Whereas the Cambridge system ignores tokens in the hypothesis that have no match in the text, my heuristic has explicit rules for matching arguments that are different. It makes my system more vulnerable to unseen cases, but at the same time makes the positive entailment decisions more well-founded. It leads my system to make fewer mistakes on the NO entailments than both the Cambridge system and SCHWA.

In their paper, Rimell & Clark do not provide an error analysis for the PETE test set, so I cannot do a comparative error analysis with my system. However, they go into detail on some analyses and mention some errors that the system made on the development data (both automatically generated and gold-standard), and I can compare these to my own results on the development data. (I will only look at those analyses where there are significant differences between Cambridge and my system.)

PETE id 5019: *He would wake up in the middle of the night and fret about it.* ⇒ *He would wake up.* The Cambridge system recognises this correctly, but the decision is based only on the single SD match `(nsubj would he)`. The other SDs are ignored, since they are non-core according to the heuristic. In my system, the YES decision is based on matching of both the relation `_would_v_modal` which has `_wake_v_up` as its scopal argument, and `_wake_v_up` itself with its pronoun argument.

PETE id 3081.N: *Occasionally, the children find steamed, whole-wheat grains for cereal which they call "buckshot".* ⇒ *Grains are steamed.* The transformation of *steamed* from an adjective in T_{sent} to a passive in H_{sent} was not accounted for in the Cambridge heuristic, and the system failed to recognise the entailment. In the MRS analyses for these sentences, *steamed* gets exactly the same representation, and my entailment system can easily match the two.

The Cambridge paper mentions that two of the errors the entailment system made were due to the fact that a non-core relation or a pronoun in the hypothesis, which Cambridge ignores, was crucial for recognising an entailment. The paper does not mention which sentences these were, but it seems likely that they would not pose a problem to my system.

4 Using 10-best MRSs

So far, I have used only one MRS per sentence in the entailment decision process. The entailment decisions were based on the best MRSs for a sentence pair, either chosen manually (treebanked MRSs) or automatically (1-best MRSs). In both cases, it can happen that the MRS chosen for a sentence is not actually the best interpretation, either because of human error during treebanking, or because the best MRS is not ranked as number one.

I also noticed that many of the incorrect decisions that the system made were caused either by errors in the MRSs or by incompatible analyses for a sentence pair. In both cases, the correct or compatible MRS could possibly be found further down the list of analyses produced by the ERG. These shortcomings can perhaps be remedied by examining more MRS analyses for each sentence in a pair.

When doing n-best parsing on the PETE data sets, we can expect a high number of analyses for the text sentences, and fewer analyses for the shorter hypotheses. By setting n to 10, I hope to capture a sufficient number of the best analyses. With 10-best parsing, I get on average 9 analyses for the text sentences, and 3 analyses for the hypotheses.

I use a simple strategy for checking entailment between a set of MRSs for the text and a set of MRSs for the hypothesis: If I can find one case of entailment between two MRSs, then I conclude that the text entails the hypothesis.

In table 4, I compare my previous results with those that I get with 10-best MRSs. As we can see, the system manages to recognise a higher number of positive entailment pairs, but the precision goes down a little. Using 10-best MRSs ensures that we do not miss out on positive entailment pairs where an incorrect MRS is ranked as number one. However, it also increases the number of spurious entailments caused by MRSs whose event relations accidentally match. Variation of *n* allows trading off precision and recall, and *n* can possibly be tuned separately for texts and hypotheses.

When we compare 10-best entailment checking to the PETE shared task results, we see that my results improve substantially over the previously highest reported performance. My system scores about 4 accuracy points higher than the system of Rimell & Clark, and more than 5 points for F1.

System	**A**	**P**	**R**	**F1**
One MRS	70.7	88.6	50.0	63.9
10-best	76.4	81.4	70.5	75.5

Table 4: Here I compare system results for one MRS and 10-best MRSs. Accuracy (A) gives the percentage of correct answers for both YES and NO. Precision (P), recall (R) and F1 are calculated for YES.

5 Conclusions and Future Work

In this paper, I have demonstrated how to build an entailment system from MRS graph alignment, combined with heuristic "robust" generalisations. I compared my results to the winner of the 2010 PETE shared task, the Cambridge system, which used grammatical relations as the basis for entailment decision. I performed an in-depth comparison of types and structure of information relevant to entailment in syntactic dependencies vs. logical-form meaning representations. The system achieved competitive results to the state of the art. Results on gold-standard parser output suggests substantially better performance in my entailment system than the PETE shared task winner.

I also generalised the approach to using n-best lists of parser outputs. Using 1-best output makes entailment decision vulnerable to incorrect MRS analyses being ranked as number one. Using n-best can counterbalance this problem. With 10-best MRSs, a significant improvement was achieved in the performance of the entailment decision system. The n-best setup offers the flexibility of trading off precision and recall.

With the 10-best MRS lists, I used a simple strategy for entailment decision: if one MRS pair supports a YES decision, we say that we have entailment. It would be interesting to explore more complex strategies, such as testing all the MRS combinations for a sentence pair for a certain *n*, and decide for the majority vote. One could also make use of the conditional probabilities on parser outputs, for instance by multiplying the probabilities for each MRS pair, summing up for YES vs. NO decisions, and setting a threshold for the final decision.

Acknowledgments

I am grateful to my supervisors Jan Tore Lønning and Stephan Oepen for suggesting this task, and for their valuable advice on my work. I also appreciate the thorough comments made by the three anonymous reviewers.

References

Richard Bergmair. 2010. *Monte Carlo Semantics: Robust Inference and Logical Pattern Processing with Natural Language Text*. Ph.D. thesis, University of Cambridge.

E. Black, S. Abney, D. Flickenger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. 1991. A Procedure for Quantitatively Comparing the Syntactic Coverage of English Grammars. In *Speech and natural language: proceedings of a workshop, held at Pacific Grove, California, February 19-22, 1991*, page 306. Morgan Kaufman Pub.

Ulrich Callmeier. 2000. PET. A platform for experimentation with efficient HPSG processing techniques. *Journal of Natural Language Engineering*, 6(1):99108, March.

John A. Carroll, Guido Minnen, and Ted Briscoe. 1999. Corpus annotation for parser evaluation. *Proceedings of the EACL workshop on Linguistically Interpreted Corpora (LINC).*

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal Recursion Semantics: An Introduction. *Research on Language & Computation*, 3(2):281–332.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL Recognising Textual Entailment Challenge. In Joaquin Quiñonero Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alché Buc, editors, *MLCW*, volume 3944 of *Lecture Notes in Computer Science*, pages 177–190. Springer.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of LREC*, Genoa, Italy.

Dan Flickinger. 2000. On building a more effcient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28.

Laura Rimell and Stephen Clark. 2010. Cambridge: Parser Evaluation using Textual Entailment by Grammatical Relation Comparison. In *Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010.*

Andreas Wotzlaw and Ravi Coote. 2013. A Logic-based Approach for Recognizing Textual Entailment Supported by Ontological Background Knowledge. *CoRR*, abs/1310.4938.

Deniz Yuret, Aydin Han, and Zehra Turgut. 2010. SemEval-2010 Task 12: Parser Evaluation using Textual Entailments. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 51–56. Association for Computational Linguistics.

A Graph-Based Approach to String Regeneration

Matic Horvat
Computer Laboratory
University of Cambridge
15 JJ Thomson Avenue, CB3 0FD, U.K.
mh693@cam.ac.uk

William Byrne
Department of Engineering
University of Cambridge
Trumpington Street, CB2 1PZ, U.K.
wjb31@cam.ac.uk

Abstract

The string regeneration problem is the problem of generating a fluent sentence from a bag of words. We explore the N-gram language model approach to string regeneration. The approach computes the highest probability permutation of the input bag of words under an N-gram language model. We describe a graph-based approach for finding the optimal permutation. The evaluation of the approach on a number of datasets yielded promising results, which were confirmed by conducting a manual evaluation study.

1 Introduction

The string regeneration problem can be stated as: given a bag of words taken from a fluent grammatical sentence, recover the original sentence. As it is often difficult to recover the exact original sentence based solely on a bag of words, the problem is relaxed to generating a fluent version of the original sentence (Zhang and Clark, 2011).

The string regeneration problem can generally be considered a difficult problem even for humans. Consider the following bag of words:

> { *Iraq, list, in, a, third, joins, the, ., of, Bush's, of, critics, policy, senator, republican* }

and try to recover the original sentence or at least a fluent grammatical sentence. The original sentence was:

> *a third republican senator joins the list of critics of Bush's policy in Iraq.*

The purpose of investigating and developing approaches to solving the string regeneration problem is grammaticality and fluency improvement of machine generated text. The output of systems generating text, including SMT, abstract-like text summarisation, question answering, and dialogue systems, often lacks grammaticality and fluency (Knight, 2007; Soricut and Marcu, 2005). The string regeneration problem is used as an application-independent method of evaluating approaches for improving grammaticality and fluency of such systems.

The string regeneration can also be viewed as a natural language realization problem. The basic task of all realization approaches is to take a meaning representation as input and generate human-readable output. The approaches differ on how much information is required from the meaning representation, ranging from semantically annotated dependency graphs to shallow syntactic dependency trees. A simple bag of words can then be considered as the least constrained input provided to a natural language realization system. The bag of words can be combined with partial constraints to form a more realistic meaning representation.

Wan et al. (2009) proposed an algorithm for grammaticality improvement based on dependency spanning trees and evaluated it on the string regeneration task. They compared its performance against a baseline N-gram language model generator. They found that their approach performs better with regards to BLEU score. The latter approach does well at a local level but nonetheless often produces ungrammatical sentences.

We argue that the authors have not fully explored the N-gram language model approach to string regeneration. They used a Viterbi-like generator with a 4-gram language model and beam pruning to find approximate solutions. Additionally, the 4-gram language model was trained on a relatively small dataset of around 20 million words.

The N-gram language model approach finds the highest probability permutation of the input bag

Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 85–95,
Gothenburg, Sweden, April 26-30 2014. ©2014 Association for Computational Linguistics

of words under an N-gram language model as the solution to the string regeneration problem. In this paper we describe a graph-based approach to computing the highest probability permutation of a bag of words. The graph-based approach models the problem as a set of vertices containing words and a set of edges between the vertices, whose cost equals language model probabilities. Finding the permutation with the highest probability in the graph formulation is equal to finding the shortest tour in the graph or, equally, solving the Travelling Salesman Problem (TSP). Despite the TSP being an NP-hard problem, state-of-the-art approaches exist to solving large problem instances. An introduction to TSP and its variants discussed in this paper can be found in Applegate et al. (2006b).

In contrast to the baseline N-gram approach by Wan et al. (2009), our approach finds optimal solutions. We built several models based on 2-gram, 3-gram, and 4-gram language models. We experimentally evaluated the graph-based approach on several datasets. The BLEU scores and example output indicated that our approach is successful in constructing a fairly fluent version of the original sentence. We confirmed the results of automatic evaluation by conducting a manual evaluation. The human judges were asked to compare the outputs of two systems and decide which is more fluent. The results are statistically significant and confirm the ranking of the systems obtained using the BLEU scores. Additionally, we explored computing approximate solutions with time constraints. We found that approximate solutions significantly decrease the quality of the output compared to optimal ones.

This paper describes work conducted in the MPhil thesis by Horvat (2013).

2 Graph-Based Approach to String Regeneration

The underlying idea of the approach discussed in this paper is to use an N-gram language model to compute the probabilities of permutations of a bag of words and pick the permutation with the highest probability as our solution.

The probability of a sequence of words under an N-gram language model is computed as:

$$\log P(w_1^n) = \sum_{k=1}^{n} \log P(w_k | w_{k-N+1}^{k-1}) \quad (1)$$

2.1 Naive Approach

A naive approach to finding the permutation with the highest probability is to enumerate all permutations, compute their probabilities using Equation 1, and choose the permutation with the highest probability as the solution.

The time complexity of the naive approach is $\mathcal{O}(n \cdot n!)$ as we are enumerating all permutations of n words and multiplying n conditional probabilities for each permutation. This means that the naive approach is not viable for sentences of even moderate length. For example, there are 3,628,800 permutations of 10 words and 355,687,428,096,000 of 17 words.

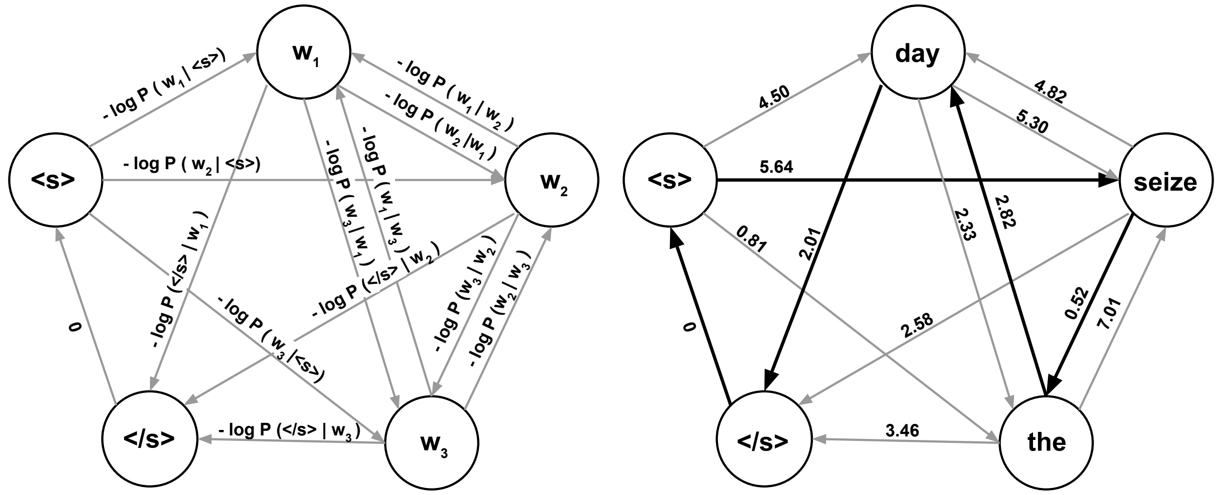
2.2 Bigram Graph-Based Approach

In this section we define the graph-based approach to finding the highest probability permutation and consequently our solution to the string regeneration problem. For a bag of words S we define a set of symbols X, $X = S \cup \{\texttt{<s>}, \texttt{</s>}\}$, which contains all the words in S (with indexes appended to distinguish repeated words) and the start and end of sentence symbols. For a bigram language model, $N = 2$, we define a directed weighted graph $G = (V, E)$, with the set of vertices defined as $V = \{w_i | w_i \in X\}$. Therefore, each symbol in X is represented by a single vertex. Let the set of edges E be a set of ordered pairs of vertices (w_i, w_j), such that $E = \{(w_i, w_j) | w_i, w_j \in V\}$. The edge cost is then defined as:

$$c_{ij} = \begin{cases} 0 & \text{if } w_i = \texttt{</s>} \\ & \text{and } w_j = \texttt{<s>}, \\ -\log P(w_j|w_i) & \text{if } w_i \neq w_j, \\ \infty & \text{otherwise.} \end{cases} \quad (2)$$

The conditional log probabilities of the form $\log P(w_j|w_i)$ are computed by a bigram language model. Consequently, finding the sentence permutation with the highest probability under the bigram language model equals finding the shortest tour in graph G or, equally, solving the Asymmetric Travelling Salesman Problem (ATSP). A general example graph for a sentence of length 3 is shown in Figure 1a.

The individual cases of the edge cost function presented in Equation 2 ensure that the solution tour is a valid sentence permutation. The negation of log probabilities transforms the problem of



(a) A general graph. The edge cost equals the negated bigram conditional log probability of the destination vertex given the origin vertex. Only edges with non-infinite edge cost are shown in the graph. Finding the shortest tour in the graph equals finding the sentence permutation with the highest probability.

(b) An example graph for the bag of words { *day, seize, the* }. The shortest tour is shown in bold and represents the word sequence *<s> seize the day </s>* with the log probability of -10.98. It is necessary to include the (</s> <s>) edge in order to complete the tour.

Figure 1: Graphs modelling a general (Figure 1a) and an example (Figure 1b) bag of words of size three under a bigram language model.

finding the longest tour in graph G to the common problem of finding the shortest tour.

Figure 1b shows a graph for the example bag of words { *day, seize, the* }. The shortest tour is shown in bold and represents the word sequence *<s> seize the day </s>*. The shortest tour equals the sentence permutation with the highest probability under the bigram language model.

The number of vertices and edges in the graph grows with the size n of the bag of words S represented by the graph $G = (V, E)$. The size of the set of vertices V in the graph is $|V| = n + 2$ and the size of the set of edges E is $|E| = |V|^2 = n^2 + 4n + 4$.

We can draw several conclusions about the graph-based approach from its equality to the TSP. Firstly, we can observe that the problem of finding the highest probability permutation is an NP-hard problem. Secondly, modelling the problem as a TSP still presents a large improvement on the naive approach described in Section 2.1. The time complexity of the naive approach for a bag of words of size n equals $\mathcal{O}(n \cdot n!)$. However, the algorithm for solving the TSP with the best-known running time guarantee has the time complexity of $\mathcal{O}(n^2 2^n)$ (Held and Karp, 1962; Applegate et al., 2006b). Although the required time grows exponentially with the length of the sentence, it grows significantly slower than with the factorial time complexity. This is illustrated in Table 1.

n	5	10	15
$n^2 2^n$	800	102,400	7,372,800
$n \cdot n!$	600	36,288,000	19,615,115,520,000

Table 1: Illustration of problem size growth at increasing values of n for algorithms with time complexity of $\mathcal{O}(n^2 2^n)$ and $\mathcal{O}(n \cdot n!)$.

Finally, by modelling the problem as a TSP we are able to take advantage of the extensive research into the TSP and choose between hundreds of algorithms for solving it. Even though no algorithm with lower running time guarantee than $\mathcal{O}(n^2 2^n)$ has been discovered since the dynamic programming algorithm described by Held and Karp (1962), many algorithms that have no guarantees but perform significantly better with most graph instances have been developed since. The size of the largest optimally solved instance has increased considerably over the years, reaching 85,900 vertices in 2006 (Applegate et al., 2009). For a more complete overview of the history and current state-of-the-art computational approaches to solving the TSP we refer the reader to Applegate et al. (2006b).

2.3 Higher N-gram Order Graph-Based Approach

Higher order N-gram language models use longer context compared to bigram language models when computing the conditional probability of the next word. This usually results in improved probability estimates for sequences of words. Therefore, to improve our initial approach using bigrams, we extend it to higher order N-grams. We first explain the intuition behind the approach and then continue with the formal definition.

The higher N-gram Order Graph-Based Approach can be modelled as a Generalized Asymmetric Travelling Salesman Problem (GATSP). GATSP is the directed (asymmetric) version of the Generalized Travelling Salesman Problem (GTSP). GTSP generalizes the TSP by grouping cities into sets called districts. GTSP can then be described as finding the shortest tour of length s, visiting exactly one city in each of the s districts. In our formulation of the graph G, each vertex has a word sequence associated with it and the districts are defined by the first word in the sequence. This means that each word appears exactly once in the solution to the GATSP, ensuring that the solution is a valid permutation. A general example graph with districts for $N = 3$ and a bag of words of size 3 is shown in Figure 2.

This is formally defined as follows. For a bag of words S we define a set of symbols X as before. For a general N-gram language model, $N > 2$, and a set of n symbols X, $|X| = n$, we define an n-partite directed weighted graph $G = (V, E)$, with the set of vertices defined as:

$$\begin{aligned} V = \{w_i | w_i[j] \in X \text{ for } 1 \leq j \leq N\text{-}1, \\ w_i[j] \neq w_i[k] \text{ for } 1 \leq j < k < N\} \end{aligned} \tag{3}$$

Each vertex therefore represents a sequence of symbols $w_i[1..N\text{-}1]$ of length $N - 1$ from the set X, and the symbols occurring in the sequence do not repeat themselves. The set of vertices V is partitioned into n disjoint independent subsets, $V_i = \{w_j | w_j \in V, w_j[1] = i\}$, based on the first word in the word sequence, $w_j[1]$.

Let the set of edges E be a set of ordered pairs of vertices (w_i, w_j), such that:

$$\begin{aligned} E = \{(w_i, w_j) | w_i \in V_k, w_j \in V_l, k \neq l, \\ w_i[2..N\text{-}1] = w_j[1..N\text{-}2], \\ w_i[1] \neq w_j[N\text{-}1]\} \end{aligned} \tag{4}$$

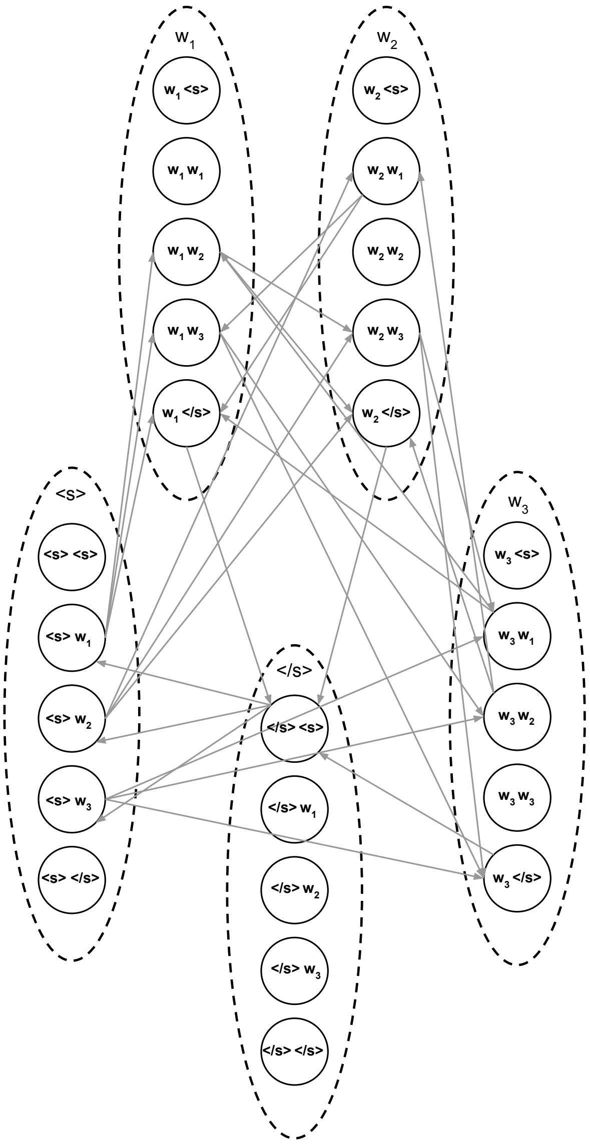


Figure 2: A general example graph for a bag of words of size 3 using a trigram language model. The graph consists of $s = 5$ districts. The vertices are assigned to a district based on the first word of the word sequence associated with the vertex. Two vertices together form a word sequence of three words. The cost of the edge between them equals the conditional probability of the final word given the context of the first two words and is provided by the trigram language model. Only edges with non-infinite cost are shown in the graph. Finding the shortest tour of length s, visiting each district exactly once, equals finding the sentence permutation with the highest probability.

$$c_{ij} = \begin{cases} 0 & \text{if } w_i[N\text{-}1] = \textit{</s>} \text{ and } w_j[N\text{-}1] = \textit{<s>}, \\ -\log P(w_j[N\text{-}1] | w_i[1..N\text{-}1]) & \text{if } w_i[k] \in S, 2 \leq k \leq N\text{-}2 \\ & \text{ and } w_i[1] \neq \textit{</s>} \text{ and } w_j[N\text{-}1] \neq \textit{<s>}, \\ -\log P(w_j[N\text{-}1] | w_i[x..N\text{-}1]) & \text{if } x \geq 2 \text{ and } w_i[x] = \textit{<s>} \\ & \text{ and } w_i[x\text{-}1] = \textit{</s>} \\ \infty & \text{otherwise.} \end{cases} \tag{5}$$

An edge therefore exists between two vertices if they are parts of two different subsets of V (have different first word in the sequence), have a matching subsequence, and the words outside the matching subsequence do not repeat between the two vertices. The edge cost is defined in Equation 5.

The conditional log probabilities of the form $\log P(w_j[N\text{-}1] | w_i[1..N\text{-}1])$ are computed by an N-gram language model. Consequently, finding the highest probability permutation under an N-gram language model equals solving the Generalized Asymmetric Travelling Salesman Problem (GATSP) modelled by graph G.

An important condition for an edge to exist between two vertices is that the word subsequences associated with the vertices match. If two vertices match, they form a word sequence of length N. The conditional log probability of the last word in the sequence given the previous $N - 1$ words equals the cost of the edge between the vertices.

An additional condition for an edge to exist between two vertices is that the words outside of the required matching subsequence do not repeat between the vertices. For example, two sequences 1 2 3 4 and 2 3 4 1 match according to the condition described above, but outside the required matching subsequence (2 3 4), word 1 appears twice which produces an invalid permutation.

The size of the vertex and edge set of the graph $G = (V, E)$ grows with the size n of the bag of words S and the order N of the N-gram language model. The size of the set of vertices V in the graph is $|V| = (n{+}2)^{N-1}$ for all values of N. The size of the set of edges E (including the infinite cost edges between the full set of vertices) is $|E| = |V|^2 = (n + 2)^{2N-2}$.

3 Implementation

The graph-based approach represents the problem of finding the sentence permutation with the highest probability as an instance of the TSP. Using a bigram language model, the problem equals solving the Asymmetric TSP. Using a higher order N-gram language model, the problem equals solving the Generalized Asymmetric TSP.

Both variations of the TSP are not as widely studied as the basic TSP and fewer algorithms exist for solving them. Transforming the variations of TSP to basic TSP is a solved problem that enables us to use state-of-the-art algorithms for solving large problem instances of the TSP. We decided to use the Concorde TSP Solver (Applegate et al., 2006a), which is prominent for continuously increasing the size of the largest optimally solved TSP instance over the last two decades.

Alternatively, a heuristic algorithm for solving the TSP can be used. Heuristic algorithms do not guarantee finding the optimal solution, but attempt to find the best possible solution given a time constraint. We used LKH as the heuristic TSP solver, which is an effective implementation of the Lin-Kernighan heuristic (Helsgaun, 2000). It currently holds the record for many large TSP instances with unknown optima.

The use of a TSP solver makes it necessary to transform the instances of ATSP and GATSP into regular TSP instances. We applied two graph transformations as necessary: (1) GATSP to ATSP transformation described by Dimitrijević and Šarić (1997) and (2) ATSP to TSP transformation described by Jonker and Volgenant (1983). These transformations allow application of the general TSP solver, although they each double the number of vertices in the graph.

Table 2 shows the total size of the vertex set after applying the transformations.

LM	5	10	15	20
2-gram	14	24	34	44
3-gram	196	576	1,156	1,936
4-gram	1,372	6,912	19,652	42,592

Table 2: The vertex set size after applying the transformations for several N-gram language models at increasing sentence length.

4 Evaluation

We evaluated three different versions of the graph-based approach based on 2-gram, 3-gram, and 4-gram language models. We evaluated each version of the system on three datasets of news sentences by computing the dataset-wide BLEU scores.

4.1 Language models

For the experimental evaluation of our graph-based approach we used 2-gram, 3-gram, and 4-gram language models with interpolated modified Kneser-Ney smoothing (Chen and Goodman, 1998). They were built using the SRI Language Modeling Toolkit (Stolcke, 2002) and KenLM Language Model Toolkit (Heafield, 2011).

The language models were estimated on the English Gigaword collection (V2, AFP and XIN parts) and the NIST OpenMT12 Evaluation Dataset (target sides of parallel data for Ar-Eng and Ch-Eng tasks).

The total size of the corpus for estimating the language models was 1.16 billion words.

4.2 Evaluation metric

The BLEU evaluation metric was developed by Papineni et al. (2002) as an inexpensive and fast method of measuring incremental progress of SMT systems. BLEU measures closeness of a candidate translation to a reference translation using N-gram precision. Similarly, in the string regeneration problem we measure the closeness of the regenerated sentence to the original sentence. We used the case insensitive NIST BLEU script v13 against tokenized references to compute the BLEU scores.

Espinosa et al. (2010) have investigated the use of various automatic evaluation metrics to measure the quality of NLG output. They found that BLEU correlates moderately well with human judgements of fluency and that it is useful for evaluation of NLG output, but should be used with caution, especially when comparing different systems. As the string regeneration problem is a basic form of NLG, BLEU is an appropriate measure of the system's performance with regards to fluency of the output. We provide examples of output and conduct a manual evaluation to confirm that the BLEU scores of individual systems reflect actual changes in output quality.

4.3 Automatic Evaluation

We evaluated the graph-based approach on three datasets:

MT08 The target side of the Ar-Eng newswire part of the NIST OpenMT08.

MT09 The target side of the Ar-Eng newswire part of the NIST OpenMT09.

SR11 The plain text news dataset of the Surface Realisation Task at GenChal'11.

The MT08, MT09, and SR11 datasets contain 813, 586, and 2398 sentences respectively.

We have taken preprocessing steps to chop long sentences into manageable parts, which is a common practice in translation. Based on preliminary experiments we decided to limit the maximum length of the chopped sentence to 20 words. N-gram models cannot be used to model sentences shorter than N words in this approach. In order to make the models comparable we ignored short sentences containing 4 or fewer words. Each chopped sentence was regenerated separately and the regenerated chopped sentences were concatenated to form the original number of dataset sentences. We expect that the preprocessing steps increased the reported BLEU scores to a certain degree. However, all systems compared in the experimental evaluation were subject to the same conditions and their scores are therefore comparable.

The graphs constructed under a 4-gram language model are too large to solve optimally in reasonable time (i.e. under half an hour per sentence). Because of this, we employ two approaches to regenerate long sentences with the 4-gram language model: (1) Use the LKH heuristic algorithm with a set time limit, and (2) back-off to the trigram language model. We refer the reader to Horvat (2013) for details.

The BLEU scores for the four systems are reported in Table 3. The 3-gram graph-based approach performed considerably better than the 2-gram approach, increasing the BLEU score for 10 BLEU points or more on all three datasets. The 4-gram approach augmented with a heuristic TSP solver performed significantly worse than the 3-gram approach on MT08 and MT09 datasets, while performing better on SR11 dataset. The reason for this difference is the different distribution of chopped sentence lengths between the three datasets. Around one fourth of all chopped

LM	Solver	MT08	MT09	SR11
2g	opt	44.4	45.1	40.6
3g	opt	57.9	58.0	50.2
4g	opt +heur	44.8	42.6	51.7
4g +3g	opt	59.1	59.5	51.8

Table 3: BLEU scores for four versions of the graph-based approach, based on 2-gram, 3-gram, and 4-gram language models. We used the 4-gram approach on sentences of up to length 18. The remaining sentences were computed using either a heuristic TSP solver (opt +heur) or by backing-off to a 3-gram approach (4g +3g).

sentences in MT08 and MT09 datasets are longer than 18 words. On the other hand, less than 1% of chopped sentences of the SR11 dataset are longer than 18 words. This means that significant parts of the MT08 and MT09 datasets were solved using the heuristic approach, compared to a small part of the SR11 dataset. Using a heuristic TSP solver therefore clearly negatively affects the performance of the system. The 4-gram approach backing-off to the 3-gram approach achieved a higher BLEU score than the 3-gram approach over all datasets.

In Figure 3 we show examples of regenerated sentences for three versions of the system. In the first example, we can see the improvements in the output fluency with better versions of the system. The improvements are reflected by the BLEU scores. The 4-gram output can be considered completely fluent. However, when compared to the original sentence, its BLEU score is not 100, due to the fact that the number of people killed and people injured are switched. In this regard, BLEU score is harsh and not an ideal evaluation metric for the task. In the second example, the original sentence contains complicated wording which is reflected in poor performance of all three versions of the system, despite the high BLEU score of the 3-gram system. In the final example, we can observe the gradual improvement of fluency over the three versions of the system. This is reflected by the BLEU score, which reaches 100.0 for the 4-gram system, which produced an identical sentence to the original.

4.4 Manual Evaluation

We manually evaluated three versions of the graph-based approach: 2-gram, 3-gram, and 4-gram system using 3-gram as back-off. We conducted a pairwise comparison of the three systems: for each evaluation sentence, we compared the output of a pair of systems and asked which output is more fluent.

We used the crowdsourcing website CrowdFlower[1] to gather fluency judgments. Judges were asked ‘Please read both sentences and compare the fluency of sentence 1 and sentence 2.’ They were given three options: ‘Sentence 1 is more fluent’, ‘Sentence 2 is more fluent’, ‘Sentence 1 and Sentence 2 are indistinguishable in fluency’. The order of presentation of the two systems was randomized for each sentence.

100 sentences of length between 5 and 18 words were chosen randomly from the combined MT08 and MT09 dataset. We gathered 5 judgements for each sentence of a single pairwise comparison of two systems. Each pairwise comparison of two systems is therefore based on 500 human judgements.

The platform measures the reliability of judges by randomly posing gold standard questions in between regular questions. If any judge incorrectly answered a number of gold standard questions, their judgements were deemed unreliable and were not used in the final result set. A thorough discussion of suitability and reliability of crowdsourcing for NLP and SMT tasks and related ethical concerns can be found in: Snow et al. (2008), Zaidan and Callison-Burch (2011), and Fort et al. (2011).

The pairwise comparison results are shown in Table 4. Each number represents the proportion of the human judgements that rated the output of the row system as better than the column system. The raw numbers of pairwise comparison judgements in favor of each system are shown in Table 5. A one-sided sign test indicated that we can reject the null hypothesis of the two systems being equal in favor of the alternative hypothesis of the first system being better than the second for all three system pairings: 3g and 2g, 4g and 2g, and 4g and 3g, $p < 0.001$ for all three comparisons. The manual evaluation results therefore confirm the BLEU score differences between the three graph-based systems.

Interestingly, in automatic evaluation the difference in BLEU scores between 2g and 3g systems was much bigger (around 10 BLEU points) than

[1] http://crowdflower.com/

	Hypothesis	**BLEU**
1. REF	meanwhile , azim stated that 10 people were killed and 94 injured in yesterday 's clashes .	
(a)	meanwhile , azim and 10 people were injured in clashes yesterday 's stated that killed 94 .	21.4
(b)	azim , meanwhile stated that 94 people were killed and 10 injured in yesterday 's clashes .	50.4
(c)	meanwhile , azim stated that 94 people were killed and 10 injured in yesterday 's clashes .	66.3
2. REF	zinni indicated in this regard that president mubarak wants egypt to work with the west .	
(a)	egypt wants zinni in this regard to work with president mubarak indicated that the west .	24.9
(b)	zinni wants egypt to work with the west that president mubarak indicated in this regard .	63.4
(c)	work with zinni indicated that president mubarak wants the west to egypt in this regard .	30.6
3. REF	he stressed that this direction is taking place in all major cities of the world .	
(a)	he stressed that the world is taking place in this direction of all major cities .	33.9
(b)	he stressed that all major cities of the world is taking place in this direction .	58.0
(c)	he stressed that this direction is taking place in all major cities of the world .	100.0

Figure 3: Output examples of three versions of the graph-based approach: (a) 2-gram, (b) 3-gram, and (c) 4-gram with 3-gram back-off. The original sentence is given for each of the three examples. Sentence BLEU scores are shown for each regenerated sentence.

LM	2g	3g	4g
2g	-	-	-
3g	65.4	-	-
4g	72.9	69.2	-

Table 4: Manual evaluation results of pairwise comparison between three versions of the system: 2-gram, 3-gram, and the 4-gram system with 3-gram back-off. The numbers represent the percentage of judgements in favor of the row system when paired with the column system.

the difference between 3g and 4g systems (around 1 BLEU point). However, in manual evaluation the difference between 3g and 4g systems is noticeably bigger (69.2%) than the difference between 2g and 3g systems (65.4%).

sys1	**sys2**	**sys1**	**equal**	**sys2**	**Total**
2g	3g	124	142	234	500
2g	4g	102	124	274	500
3g	4g	92	201	207	500

Table 5: The raw numbers of pairwise comparison judgements between the three systems. The columns give the number of judgements in favor of each of the three options.

5 Related Work

The basic task of all natural language realization approaches is to take a meaning representation as input and generate human-readable output. The approaches differ on how much information is required from the meaning representation. Deep representation include dependency graphs annotated with semantic labels and other syntactic information (Belz et al., 2011). Shallow representations include syntactic dependency trees annotated with POS tags and other syntactic information (Belz et al., 2011), IDL-expressions (Soricut and Marcu, 2005), and Abstract Meaning Representation (Langkilde and Knight, 1998).

Soricut and Marcu (2005) consider NLG in context of other popular natural language applications, such as Machine Translation, Summarization, and Question Answering. They view these as text-to-text applications that produce textual output from textual input. Because of this, many natural language applications need to include some form of natural language generation to produce the output text. However, the natural language generation in these applications is often handled in an application-specific way. They propose to use IDL-expressions as an application-independent representation language for text-to-text NLG. The IDL-expressions are created from strings using operators to combine them. The authors evaluate their approach on the string regeneration task and achieve moderate BLEU scores.

Wan et al. (2009) approach the string regeneration problem using dependency spanning trees. Their approach is to search for the most probable dependency tree containing each word in the input or, equally, finding the optimal spanning tree. Zhang and Clark (2011) propose a similar approach using Combinatory Categorial Grammar (CCG) which imposes stronger category constraints on the parse structure compared to dependency trees investigated by Wan et al. (2009). They primarily focus on the search problem of finding an optimal parse tree among all possible trees containing any choice and ordering of the input words. The CCG approach achieved higher BLEU scores compared to the approach proposed by Wan et al. (2009). Zhang et al. (2012) improve the CCG approach by Zhang and Clark (2011) by incorporating an N-gram language model. de Gispert et al. (2014) present a similar N-gram language model approach to ours with a different decoder that does not guarantee optimal results. In their comparison with approach by Zhang et al. (2012) they report gains of more than 20 BLEU points.

The purpose of studying and building approaches to solving the string regeneration problem is to improve grammaticality and fluency of machine generated text. An approach using a TSP reordering model by Visweswariah et al. (2011) focused on the preordering task in SMT. In the preordering task the words of the source sentence are reordered to reflect the word order expected in the target sentence which helps improve the performance of the SMT system.

6 Conclusions and Future Work

In the paper we explored the N-gram language model approach to the string regeneration problem of recovering a fluent version of the original sentence given a bag of words. The N-gram language model approach computes the highest probability permutation of the input bag of words under an N-gram language model. We described a graph-based approach for finding the optimal permutation. Finding the permutation with the highest probability in the graph formulation is equal to finding the shortest tour in the graph or, equally, solving the Travelling Salesman Problem.

We evaluated the proposed approach on three datasets. The BLEU scores and example output indicated that the graph-based approach is successful in constructing a fairly fluent version of the original sentence. The 2-gram based approach performed moderately well but was surpassed by the 3-gram based approach. The 4-gram based approach offered an improvement on the 3-gram but is not of much practical use due to its long computation times. Approximate solutions computed using a heuristic TSP solver significantly reduced the quality of the output and resulting BLEU score. We confirmed the results of automatic evaluation by conducting a manual evaluation.

The BLEU scores of our approach and the approach by Wan et al. (2009) can't be directly compared as we used different evaluation datasets and preprocessing procedures. Nonetheless, the difference in BLEU scores is stark, our best system outperforming theirs by more than 20 BLEU points.

The work presented in this paper can be extended in a number of ways. More extensive comparison between optimal and approximate approaches would help draw stronger conclusions regarding the need for optimality. A direct comparison between our N-gram language model based approach and approaches presented by Wan et al. (2009), Zhang et al. (2012), and others is needed to determine its performance relative to other approaches.

The graph-based approach itself can be extended in a number of ways. Emulating methods from Statistical Machine Translation, the approach could be extended to generate an N-best list of reorderings. A different method could then be used to rerank the N-best list to choose the best one. The methods can range from rescoring the outputs with a higher-order language model or a dependency language model, to using discriminative machine learning. The approach could also be extended to handle additional constraints in the input, such as phrases instead of words, by modifying the edge weights of the graph.

Another interesting area of future research relating to the wider string regeneration problem is determining the human performance on the task. Based on a simple trial of trying to regenerate a long sentence by hand, it is clear that human performance on the task would not equal 100 BLEU points. It would therefore be interesting to determine the human performance on the string regeneration problem to provide a contrast and a point of comparison to the performance of machine systems.

Finally, the string regeneration problem can be viewed as a constraint satisfaction approach where the constraints are minimal. However, in many instances there is more information available regarding the final output of a system, for example syntactic or semantic relationship between words. This information introduces additional constraints to the simple bag of words that need to be included in the output. In future, we will explore methods of generating from a set of constraints in a robust manner to produce output that is fluent and grammatical.

Acknowledgments

We thank Juan Pino for his help with language modeling and experiments. We thank Ann Copestake and the anonymous reviewers for their discussions and suggestions.

Matic Horvat was supported by the Ad Futura Scholarship during his work on the MPhil thesis.

References

David L. Applegate, Robert E. Bixby, Vašek Chvátal, and William J. Cook. 2006a. Concorde TSP Solver.

David L. Applegate, Robert E. Bixby, Vašek Chvátal, and William J. Cook. 2006b. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press.

David L. Applegate, Robert E. Bixby, Vašek Chvátal, William Cook, Daniel G. Espinoza, Marcos Goycoolea, and Keld Helsgaun. 2009. Certification of an optimal TSP tour through 85,900 cities. *Operations Research Letters*, 37(1):11–15, January.

Anja Belz, Michael White, Dominic Espinosa, Eric Kow, Deirdre Hogan, and Amanda Stent. 2011. The First Surface Realisation Shared Task : Overview and Evaluation Results. In *Proceedings of the 13th European Workshop on Natural Language Generation (ENLG)*, volume 2, pages 217–226.

Stanley F. Chen and Joshua T. Goodman. 1998. An Empirical Study of Smoothing Techniques for Language Modeling. Technical report, Harvard University.

Adrià de Gispert, Marcus Tomalin, and William Byrne. 2014. Word Ordering with Phrase-Based Grammars. In *Proceedings of the European Chapter of the Association for Computational Linguistics*, number 2009.

Vladimir Dimitrijević and Zoran Šarić. 1997. An Efficient Transformation of the Generalized Traveling Salesman Problem into the Traveling Salesman Problem on DIgraphs. *Information Sciences*, 102:105–110.

Dominic Espinosa, Rajakrishnan Rajkumar, Michael White, and Shoshana Berleant. 2010. Further Meta-Evaluation of Broad-Coverage Surface Realization. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 564–574.

Karën Fort, Adda Gilles, and K. Bretonnel Cohen. 2011. Amazon Mechanical Turk: Gold Mine or Coal Mine? *Computational Linguistics*, 37(2):413–420.

Kenneth Heafield. 2011. KenLM : Faster and Smaller Language Model Queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, number 2009, pages 187–197.

Michael Held and Richard M. Karp. 1962. A Dynamic Programming Approach to Sequencing Problems. *Society for Industrial and Applied Mathematics*, 10(1):196–210.

Keld Helsgaun. 2000. An effective implementation of the LinKernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1):106–130.

Matic Horvat. 2013. *A Graph-Based Approach to String Regeneration*. Ph.D. thesis.

Roy Jonker and Ton Volgenant. 1983. Transforming Asymmetric into Symmetric Traveling Salesman Problems. *Operations Research Letters*, 2(4):161–163.

Kevin Knight. 2007. Automatic language translation generation help needs badly. In *MT Summit XI Work- shop on Using Corpora for NLG: Keynote Address*, pages 5–8.

Irene Langkilde and Kevin Knight. 1998. Generation that Exploits Corpus-Based Statistical Knowledge. In *Proceedings of the 17th international conference on Computational linguistics*, pages 704–710.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-jing Zhu. 2002. BLEU : a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, number July, pages 311–318, Philadelphia.

Rion Snow, Brendan O Connor, Daniel Jurafsky, and Andrew Y Ng. 2008. Cheap and Fast But is it Good ? Evaluating Non-Expert Annotations for Natural Language Tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, number October, pages 254–263.

Radu Soricut and Daniel Marcu. 2005. Towards Developing Generation Algorithms for Text-to-Text Applications. In *Proceedings of the 43rd Annual Meeting of the ACL*, number June, pages 66–74, Ann Arbor.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, pages 901–904.

Karthik Visweswariah, Rajakrishnan Rajkumar, Ankur Gandhe, Ananthakrishnan Ramanathan, and Jiri Navratil. 2011. A Word Reordering Model for Improved Machine Translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 486–496.

Stephen Wan, Mark Dras, Robert Dale, and Cecile Paris. 2009. Improving Grammaticality in Statistical Sentence Generation : Introducing a Dependency Spanning Tree Algorithm with an Argument Satisfaction Model. In *Proceedings of the 12th Conference of the European Chapter of the ACL*, number April, pages 852–860.

Omar F Zaidan and Chris Callison-Burch. 2011. Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1220–1229.

Yue Zhang and Stephen Clark. 2011. Syntax-Based Grammaticality Improvement using CCG and Guided Search. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1147–1157.

Yue Zhang, Graeme Blackwood, and Stephen Clark. 2012. Syntax-Based Word Ordering Incorporating a Large-Scale Language Model. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 736–746, Avignon, France.

Complexity of Word Collocation Networks: A Preliminary Structural Analysis

Shibamouli Lahiri
Computer Science and Engineering
University of North Texas
Denton, TX 76207, USA
shibamoulilahiri@my.unt.edu

Abstract

In this paper, we explore complex network properties of word collocation networks (Ferret, 2002) from four different genres. Each document of a particular genre was converted into a network of words with word collocations as edges. We analyzed graphically and statistically how the *global properties* of these networks varied across different genres, and among different network types within the same genre. Our results indicate that the distributions of network properties are visually similar but statistically apart across different genres, and interesting variations emerge when we consider different network types within a single genre. We further investigate how the global properties change as we add more and more collocation edges to the graph of one particular genre, and observe that except for the number of vertices and the size of the largest connected component, network properties change in *phases*, via jumps and drops.

1 Introduction

Word collocation networks (Ferret, 2002; Ke, 2007), also known as collocation graphs (Heyer et al., 2001; Choudhury and Mukherjee, 2009), are networks of words found in a document or a document collection, where each node corresponds to a unique *word type*, and edges correspond to *word collocations* (Ke and Yao, 2008). In the simplest case, each edge corresponds to a unique bigram in the original document. For example, if the words w_A and w_B appeared together in a document as a bigram $w_A w_B$, then the word collocation network of that particular document will contain an edge $w_A \rightarrow w_B$. Note that edges can be directed ($w_A \rightarrow w_B$) or undirected ($w_A - w_B$). Furthermore, they can be weighted (with the frequency of the bigram $w_A w_B$) or unweighted.

It is interesting to note that word collocation networks display complex network structure, including power-law degree distribution and small-world behavior (Matsuo et al., 2001a; Matsuo et al., 2001b; Masucci and Rodgers, 2006; Liang et al., 2012). This is not surprising, given that natural language generally shows complex network properties at different levels (Ferrer i Cancho and Solé, 2001; Motter et al., 2003; Biemann et al., 2009; Liang et al., 2009). Moreover, researchers have used such complex networks in applications ranging from text genre identification (Stevanak et al., 2010) and Web query analysis (Saha Roy et al., 2011) to semantic analysis (Biemann et al., 2012) and opinion mining (Amancio et al., 2011). In Section 2, we will discuss some of these applications in more detail.

The goal of this paper is to explore some key structural properties of these complex networks (cf. Table 1), and study how they vary across different genres of text, and also across different network types within the same genre. We chose *global network properties* like diameter, global clustering coefficient, shrinkage exponent (Leskovec et al., 2007), and small-worldliness (Walsh, 1999; Matsuo et al., 2001a), and experimented with four different text collections – blogs, news articles, academic papers, and digitized books (Section 4.1). Six different types of word collocation networks were constructed on each document, as well as on the entire collections – two with directed edges, and four with undirected edges (Section 3). We did not take into account edge weights in our study, and kept it as a part of our future work (Section 5).

Tracking the variation of complex network properties on word collocation networks yielded several important observations and insights. We

Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 96–105,
Gothenburg, Sweden, April 26-30 2014. ©2014 Association for Computational Linguistics

noted in particular that different genres had considerable visual overlap in the distributions of global network properties like diameter and clustering coefficient (cf. Figure 2), although statistical significance tests indicated the distributions were sufficiently apart from each other (Section 4.2). This calls for a deeper analysis of complex network properties and their general applicability to tasks like genre identification (Stevanak et al., 2010).

We further analyzed distributions of global word network properties across six different network types *within the same genre* (Section 4.2). This time, however, we noted a significant amount of separation – both visually as well as statistically – among the distributions of different global properties (cf. Figure 3 and Table 5).

In our final set of experiments, we analyzed how global network properties change as we start with an empty network, and gradually add edges to that network. For this experiment, we chose the news genre, and tracked the variation of 17 different global network properties on four types of networks. We observed that all global network properties (except the number of vertices and edges, number of connected components and the size of the largest connected component) show unpredictability and *spikes* when the percentage of added edges is small. We also noted that most global properties showed at least one *phase transition* as the word collocation networks grew larger. Statistical significance tests indicated that the patterns of most global property variations were nonrandom and positively correlated (Section 4.3).

2 Related Work

That language shows complex network structure at the word level, was shown more than a decade ago by at least two independent groups of researchers (Ferrer i Cancho and Solé, 2001; Matsuo et al., 2001a). Matsuo et al. (2001b) went further ahead, and designed an unsupervised keyword extraction algorithm using the small-world property of word collocation networks. Motter et al. (2003) extended the collocation network idea to *concepts* rather than words, and observed a small-world structure in the resulting network. Edges between concepts were defined as entries in an English thesaurus. Liang et al. (2009) compared word collocation networks of Chinese and English text, and pointed out their similarities and differences. They further constructed *character collocation networks* in Chinese, showed their small-world structure, and used these networks in a follow-up study to accurately segregate Chinese essays from different literary periods (Liang et al., 2012).

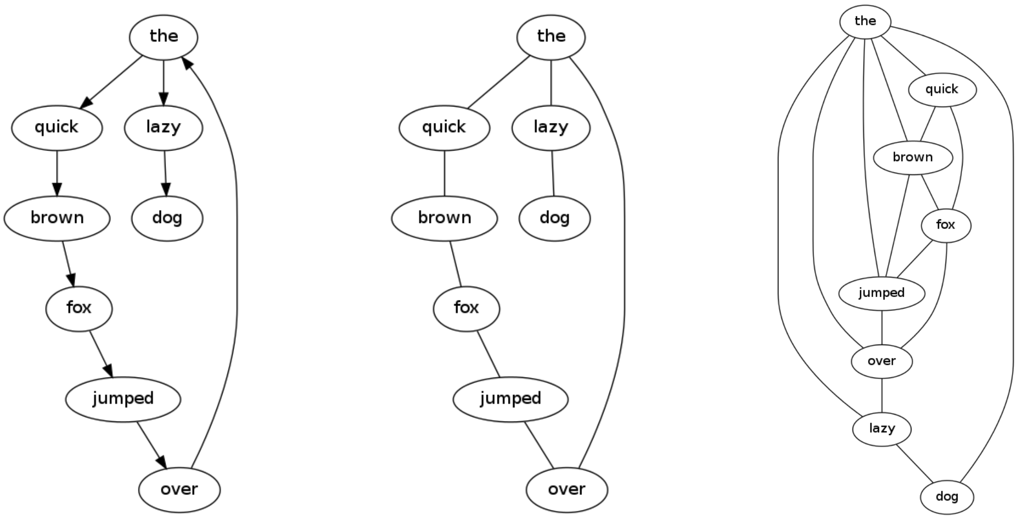
Word collocation networks have also been successfully applied to the authorship attribution task.[1] Antiqueira et al. (2006) were among the first to apply complex network features like clustering coefficient, *component dynamics deviation* and *degree correlation* to the authorship attribution problem.

Biemann et al. (2009) constructed syntactic and semantic distributional similarity networks (DSNs), and analyzed their structural differences using spectral plots. Biemann et al. (2012) further used *graph motifs* on collocation networks to distinguish real natural language text from generated natural language text, and to point out the shortcomings of n-gram language models.

Word collocation networks have been used by Amancio et al. (2011) for opinion mining, and by Mihalcea and Tarau (2004) for keyword extraction. While the former study used complex network properties as features for machine learning algorithms, the latter ran PageRank (Page et al., 1998) on word collocation networks to sieve out most important words.

While all the above studies are very important, we found none that performed a thorough and systematic exploration of different global network properties on different network types across genres, along with statistical significance tests to assess the validity of their observations. Stevanak et al. (2010), for example, used word collocation networks to distinguish between novels and news articles, but they did not perform a distributional analysis of the different global network properties they used, thereby leaving open how good those properties truly were as features for genre classification, and whether there exist a better and simpler set of global network properties for the same task. On the other hand, Masucci and Rodgers (2006), Ke (2007), and Ke and Yao (2008) explored several global network properties on word collocation networks, but they did not address the problem of analyzing within-genre and cross-genre variations of those properties.

[1] For details on authorship attribution, please see the surveys by Juola (2006), Koppel et al. (2009), and Stamatatos (2009).



(a) Directed (b) Undirected, Variant 1 (c) Undirected, Variant 2

Figure 1: Word collocation networks of the sentence *"the quick brown fox jumped over the lazy dog"*. Note that for all three network types, the word "the" appeared as the most central word. It is in general the case that stop words like "the" are the most central words in collocation networks, especially since they act as *connectors* between other words.

Network Property	**Mathematical Expression**
Number of vertices	$\|V\|$
Number of edges	$\|E\|$
Shrinkage exponent (Leskovec et al., 2007)	$\log_{\|V\|} \|E\|$
Global clustering coefficient	C
Small-worldliness (Walsh, 1999; Matsuo et al., 2001a)	$\mu = (\bar{C}/L)/(\bar{C}_{rand}/L_{rand})$
Diameter (directed)	d
Diameter (undirected)	d
Power-law exponent of degree distribution	α
Power-law exponent of in-degree distribution	α_{in}
Power-law exponent of out-degree distribution	α_{out}
p-value for the power-law exponent of degree distribution	N/A
p-value for the power-law exponent of in-degree distribution	N/A
p-value for the power-law exponent of out-degree distribution	N/A
Number of connected components*	N/A
Size of the largest connected component*	N/A
Number of strongly connected components*	N/A
Size of the largest strongly connected component*	N/A

Table 1: Different global network properties used in our study. The ones marked with an asterisk ("*") are only used in Section 4.3 in the context of incrementally constructing networks by gradually adding edges. For document networks, these four properties do not make sense, because the number of connected components is always one, and the size of the largest connected component always equals the number of vertices in the document network. Note also that in-degree distribution, out-degree distribution, and the directed version of diameter do not make sense for undirected networks, and same goes with the number of strongly connected components and the size of the largest strongly connected component. Here we report them separately for conceptual clarity.

In addition to addressing these problems, in this paper we introduce a new analysis - how the global network properties change as we gradually add more collocation edges to a network (Section 4.3).[2]

3 Collocation Networks of Words

Before constructing collocation networks, we lowercased the input text and removed all punctuation, but refrained from performing stemming in order to retain subtle distinctions between words like "vector" and "vectorization". Six different types of word collocation networks were constructed on each document (used in Section 4.2) as well as on document collections (used in Section 4.3), where nodes are unique words, and an edge appears between two nodes if their corresponding words appeared together as a bigram or in a trigram in the original text. All the network types have the *same* number of vertices (i.e., words) for a particular document or a document collection, and they are only distinguished from each other by the type (and potentially, number) of edges, as follows:

Directed – Directed edge $w_A \rightarrow w_B$ if $w_A w_B$ is a bigram in the given text.

Undirected, Variant 1 – Undirected edge $w_A - w_B$ if $w_A w_B$ is a bigram in the given text.

Undirected, Variant 2 – Undirected edges $w_A - w_B$, $w_B - w_C$ and $w_A - w_C$, if $w_A w_B w_C$ is a trigram in the given text.

Directed Simplified – Same as the directed version, with *self-loops* removed.[3]

Undirected Variant 1, Simplified – Same as the undirected variant 1, with self-loops removed.

Undirected Variant 2, Simplified – Same as the undirected variant 2, with self-loops removed.

We did not take into account edge weights in our study, and all our networks are therefore unweighted networks. Furthermore, since we removed all punctuation information *before* constructing collocation networks, sentence boundaries were implicitly ignored. In other words, the last word of a sentence *does* link to the first word of the next sentence in our collocation networks. An example of the first three types of networks (directed, undirected variant 1, and undirected variant 2) is shown in Figure 1. Here we considered a sentence *"the quick brown fox jumped over the lazy dog"* as our document. Note that all the collocation networks in Figure 1 contain at least one cycle, and the directed version contains a directed cycle. In a realistic document network, there can be many such cycles.

We constructed word collocation networks on *document collections* as well. In this case, the six network types remain as before, and the only difference comes from the fact that now the whole collection is considered a single *super-document*. Words in this super-document are connected according to bigram and trigram relationships. We respected document boundaries in this case, so the last word of a particular document *does not* link to the first word of the next document. The *collection networks* have only been used in Section 4.3 of this paper, to show how global network properties change as we add edges to the network.

With the networks now constructed, we went ahead and explored several of their global properties (cf. Table 1). Properties were measured on each type of network on each document, thereby giving us property distributions across different genres of documents for a particular network type (cf. Figure 2), as well as property distributions across different network types for a particular genre (cf. Figure 3). We used the *igraph* software package (Csardi and Nepusz, 2006) for computing global network properties.

Among the properties in Table 1, number of vertices ($|V|$) and number of edges ($|E|$) are self-explanatory. The *shrinkage exponent* ($\log_{|V|} |E|$) is motivated by the observations that the number of edges ($|E|$) follows a power-law relationship with the number of vertices ($|V|$), and that as a network evolves, both $|V|$ and $|E|$ continue to grow, but the diameter of the network either *shrinks* or plateaus out, thereby resulting in a *densified* network (Leskovec et al., 2007). We explored two versions of graph diameter (d) in our study - a directed version (considering directed edges), and an undirected version (ignoring edge directions).[4]

The *global clustering coefficient* (C) is a mea-

[2] All code, data, and supplementary material are available at `https://drive.google.com/file/d/0B2Mzhc7popBgODFKZVVnQTFMQkE/edit?usp=sharing`. The data includes – among other things – the corpora we used (cf. Section 4.1), and code to construct the networks and analyze their properties.

[3] Note that self-loops may appear in word collocation networks due to punctuation removal in the pre-processing step. An example of such a self-loop is: *"The airplane took off. Off we go to Alaska."* Here the word "off" will contain a self-loop.

[4] For undirected collocation networks, these two versions yield the same results, as expected.

sure of how interconnected a graph's nodes are among themselves. It is defined as the ratio between the number of closed triplets of vertices (i.e., the number of ordered triangles or *transitive triads*), and the number of connected vertex-triples (Wasserman and Faust, 1994). The *small-worldliness* or *proximity ratio* (μ) of a network measures to what extent the network exhibits small-world behavior. It is quantified as the amount of deviation of the network from an equally large random network, in terms of average local clustering coefficient ($\bar{C}$) and average shortest path length (L)[5]. The exact ratio is $\mu = (\bar{C}/L)/(\bar{C}_{rand}/L_{rand})$, where $\bar{C}$ and L are the average local clustering coefficient and the average shortest path length of the given network, and $\bar{C}_{rand}$ and L_{rand} are the average local clustering coefficient and the average shortest path length of an equally large random network (Walsh, 1999; Matsuo et al., 2001a).

Since collocation networks have been found to display scale-free (power-law) degree distribution in several previous studies (see, e.g., (Ferrer i Cancho and Solé, 2001; Masucci and Rodgers, 2006; Liang et al., 2009)), we computed power-law exponents of in-degree, out-degree, and degree distributions on each of our collocation networks.[6] We also computed the corresponding p-values, following a procedure outlined in (Clauset et al., 2009). These p-values help assess whether the distributions are power-law or not. If a p-value is < 0.05, then there is statistical evidence to believe that the corresponding distribution is *not* a power-law distribution.

Finally, we computed the number of connected components, size of the largest ("giant") connected component, number of strongly connected components, and size of the largest strongly connected component, to be used in Section 4.3.

4 Analysis of Network Properties

4.1 Datasets

We used four document collections from four different genres – blogs, news articles, academic papers, and digitized books. For blogs, we used the **Blog Authorship Corpus** created by (Schler et al., 2006). It consists of 19,320 blogs from authors of different age groups and professions. The unprocessed corpus has about 136.8 million word tokens.

Our news articles come from the **Reuters-21578, Distribution 1.0** collection.[7] This collection contains 19,043 news stories, and about 2.6 million word tokens (unprocessed).

For the academic paper dataset, we used **NIPS Conference Papers Vols 0-12**.[8] This corpus comprises 1,740 papers and about 4.8 million unprocessed word tokens.

Finally, we created our own corpus of 3,036 digitized books written by 142 authors from the **Project Gutenberg** digital library.[9] After removing metadata, license information, and transcribers' notes, this dataset contains about 210.9 million word tokens.

That the word collocation networks of individual documents are indeed scale-free and small-world, is evident from Tables 2, 3, and 4, and Figure 2h. Irrespective of network type, a majority of the median α (power-law exponent of degree distribution) values hovers in the range $[2, 3]$, with low dispersion. This corroborates with earlier studies (Ferrer i Cancho and Solé, 2001; Liang et al., 2009; Liang et al., 2012). Similarly, the median μ (small-worldliness) is high for all genres except *news* (irrespective of network type), thereby indicating the document networks are indeed small-world. This finding is in line with previous studies (Matsuo et al., 2001a; Matsuo et al., 2001b). Moreover, Figure 2h shows that a majority of documents in different genres have a very high p-value, indicating that the networks are significantly power-law. The *news* genre poses an interesting case. Since many news stories in the Reuters-21578 collection are small, their collocation networks are not very well-connected, thereby resulting in very low small-worldliness values, as well as higher estimates of the power-law exponent α (cf. Tables 2, 3, and 4).

4.2 Distribution of Global Network Properties

We plotted the histograms of eight important global network properties on directed collocation networks in Figure 2. All histograms were plot-

[5] Also called *"characteristic path length"* (Watts and Strogatz, 1998).

[6] For undirected graphs, the exponents on all three distributions are the same.

[7] Available from `http://www.daviddlewis.com/resources/testcollections/reuters21578/`.

[8] Available from `http://www.cs.nyu.edu/~roweis/data.html`.

[9] `http://www.gutenberg.org/`.

Dataset	**Median α on Digraph**	**Median α on Undigraph 1**	**Median α on Undigraph 2**	**Median μ on Digraph**	**Median μ on Undigraph 1**	**Median μ on Undigraph 2**
	(quartile deviations are in parentheses)			(quartile deviations are in parentheses)		
Blog	2.34 (0.17)	2.34 (0.17)	2.41 (0.19)	16.63 (17.16)	22.50 (22.01)	14.93 (9.49)
News	3.38 (0.42)	3.38 (0.42)	4.35 (0.98)	0.63 (0.50)	0.95 (0.76)	1.75 (0.71)
Papers	2.35 (0.09)	2.35 (0.09)	2.45 (0.11)	20.69 (2.96)	27.87 (3.93)	14.95 (1.80)
Digitized Books	2.12 (0.04)	2.12 (0.04)	2.16 (0.05)	244.31 (98.62)	296.73 (116.98)	88.46 (31.78)
All together	2.58 (0.53)	2.58 (0.53)	2.70 (0.90)	5.03 (11.93)	7.27 (15.85)	7.31 (8.47)

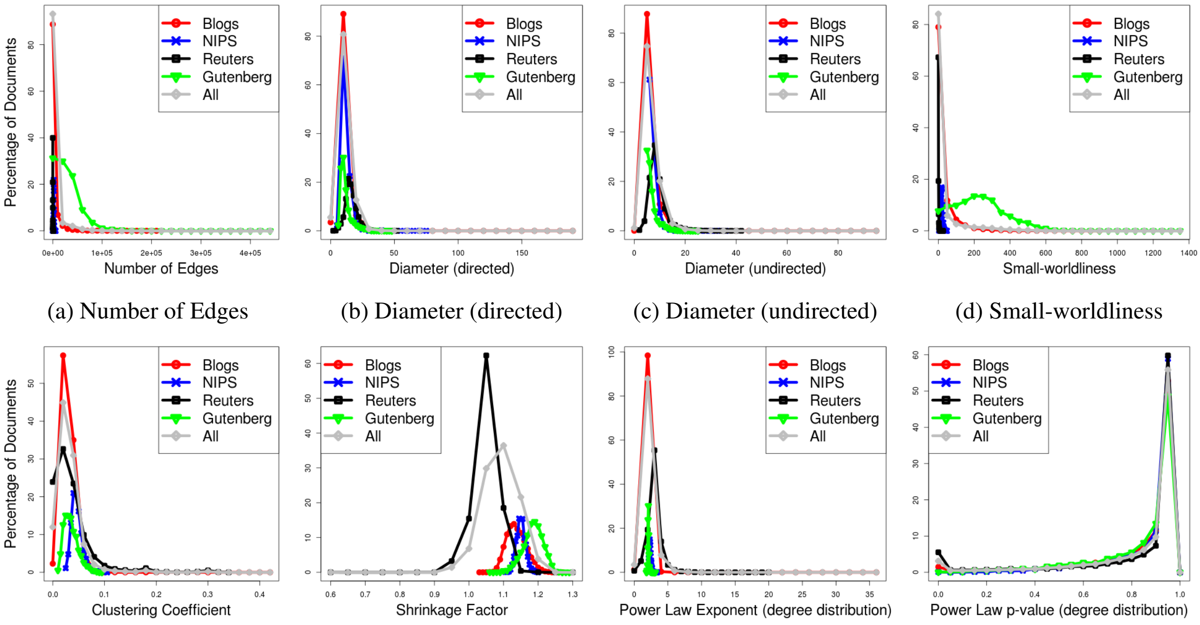
Table 2: Power-law exponent of degree distribution (α) and small-worldliness (μ) of word collocation networks. Here we report the median across documents in a particular dataset (genre), and also the median across all documents in all datasets (last row).

Dataset	**Median α on Simplified Digraph**	**Median α on Simplified Undigraph 1**	**Median α on Simplified Undigraph 2**	**Median μ on Simplified Digraph**	**Median μ on Simplified Undigraph 1**	**Median μ on Simplified Undigraph 2**
	(quartile deviations are in parentheses)			(quartile deviations are in parentheses)		
Blog	2.34 (0.17)	2.34 (0.16)	2.36 (0.18)	16.67 (17.18)	23.28 (22.98)	39.13 (24.03)
News	3.39 (0.42)	3.40 (0.42)	3.88 (0.77)	0.63 (0.50)	0.96 (0.77)	4.96 (1.93)
Papers	2.36 (0.09)	2.37 (0.09)	2.40 (0.11)	20.78 (2.98)	29.18 (4.09)	38.81 (4.75)
Digitized Books	2.12 (0.04)	2.13 (0.04)	2.14 (0.05)	244.53 (98.81)	317.49 (127.14)	218.77 (78.02)
All together	2.58 (0.53)	2.58 (0.54)	2.65 (0.72)	5.04 (11.97)	7.45 (16.52)	19.64 (21.82)

Table 3: Power-law exponent of degree distribution (α) and small-worldliness (μ) of word collocation networks. Here we report the median across documents in a particular dataset (genre), and also the median across all documents in all datasets (last row).

Network Type	**Median α on Blogs**	**Median α on Papers**	**Median α on News**	**Median α on Books**	**Median α on All**	**Median μ on Blogs**	**Median μ on Papers**	**Median μ on News**	**Median μ on Books**	**Median μ on All**
	(quartile deviations are in parentheses)					(quartile deviations are in parentheses)				
Digraph	2.34 (0.17)	2.35 (0.09)	3.38 (0.42)	2.12 (0.04)	2.58 (0.53)	16.63 (17.16)	20.69 (2.96)	0.63 (0.50)	244.31 (98.62)	5.03 (11.93)
Undigraph 1	2.34 (0.17)	2.35 (0.09)	3.38 (0.42)	2.12 (0.04)	2.58 (0.53)	22.50 (22.01)	27.87 (3.93)	0.95 (0.76)	296.73 (116.98)	7.27 (15.85)
Undigraph 2	2.41 (0.19)	2.45 (0.11)	4.35 (0.98)	2.16 (0.05)	2.70 (0.90)	14.93 (9.49)	14.95 (1.80)	1.75 (0.71)	88.46 (31.78)	7.31 (8.47)
Simplified Digraph	2.34 (0.17)	2.36 (0.09)	3.39 (0.42)	2.12 (0.04)	2.58 (0.53)	16.67 (17.18)	20.78 (2.98)	0.63 (0.50)	244.53 (98.81)	5.04 (11.97)
Simplified Undigraph 1	2.34 (0.16)	2.37 (0.09)	3.40 (0.42)	2.13 (0.04)	2.58 (0.54)	23.28 (22.98)	29.18 (4.09)	0.96 (0.77)	317.49 (127.14)	7.45 (16.52)
Simplified Undigraph 2	2.36 (0.18)	2.40 (0.11)	3.88 (0.77)	2.14 (0.05)	2.65 (0.72)	39.13 (24.03)	38.81 (4.75)	4.96 (1.93)	218.77 (78.02)	19.64 (21.82)

Table 4: Power-law exponent of degree distribution (α) and small-worldliness (μ) of word collocation networks. Here we report the median across documents for a particular network type.



(a) Number of Edges

(b) Diameter (directed)

(c) Diameter (undirected)

(d) Small-worldliness

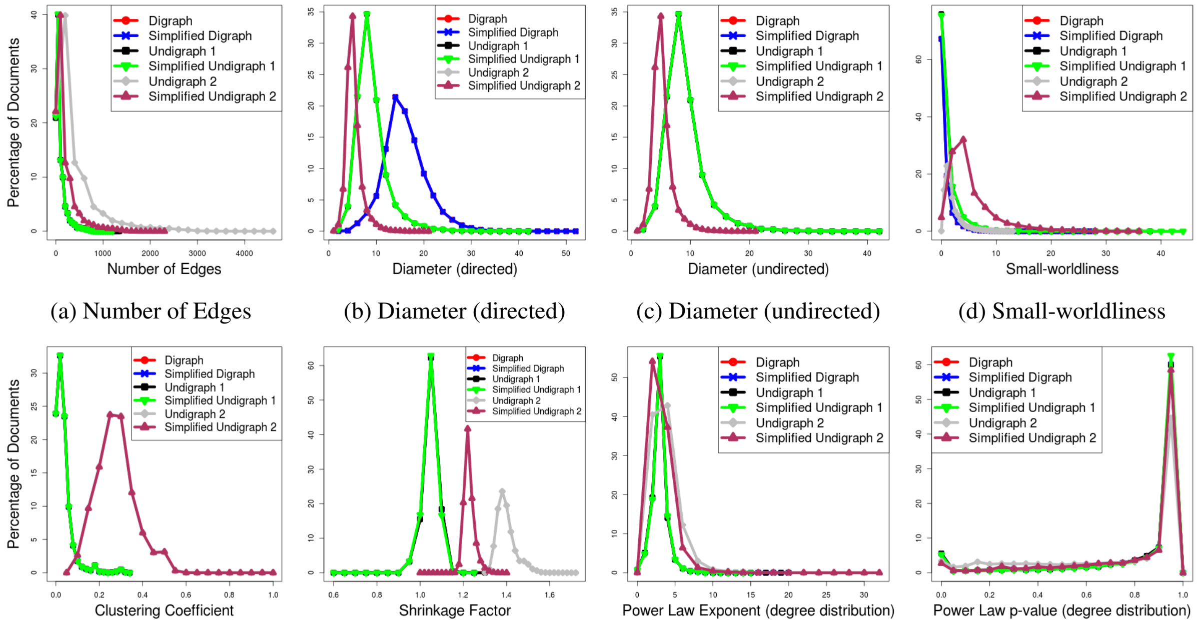
(e) Clustering Coefficient

(f) Shrinkage Exponent

(g) α

(h) p-value for α

Figure 2: Distributions of eight global network properties across different genres for directed collocation networks. Y-axes represent the percentage of documents for different genres.



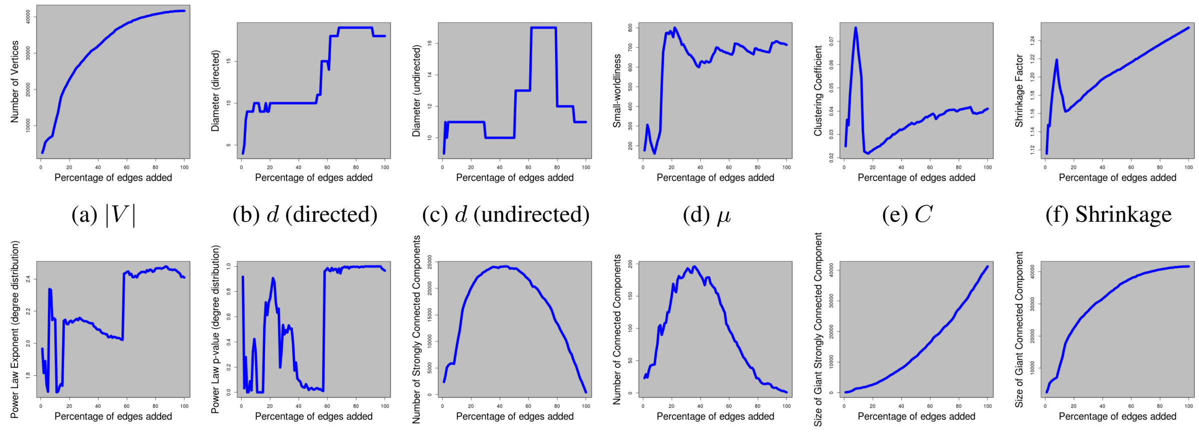
(a) Number of Edges (b) Diameter (directed) (c) Diameter (undirected) (d) Small-worldliness

(e) Clustering Coefficient (f) Shrinkage Exponent (g) α (h) p-value for α

Figure 3: Distributions of eight global network properties across different network types on the *news* genre. Y-axes represent the percentage of documents for different network types.

Test	$\|E\|$	**Directed** d	**Undirected** d	μ	C	**Shrinkage**	α	**p-value for** α
ANOVA	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001
Kruskal-Wallis	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001
ANOVA	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001
Kruskal-Wallis	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001

Table 5: p-values from ANOVA and Kruskal-Wallis tests. The top two rows are p-values for Figure 2, and the bottom two rows are p-values for Figure 3. Each column corresponds to one subfigure of Figure 2 and Figure 3. p-values in general were extremely low - close to zero in most cases.



(a) $|V|$ (b) d (directed) (c) d (undirected) (d) μ (e) C (f) Shrinkage

(g) α (h) p-value for α (i) Number of SCCs (j) Number of CCs (k) Giant SCC Size (l) Giant CC Size

Figure 4: Change of global network properties with incremental addition of edges to the directed network of *news* genre. SCC = Strongly Connected Component, CC = Connected Component. By "giant" CC and "giant" SCC, we mean the largest CC and the largest SCC. See Table 1 for other properties.

ted with 20 bins. Figure 2e, for example, shows the global clustering coefficient (C) on the X-axis, divided into 20 bins, and the percentage of document networks (directed) with C values falling into a particular bin, on the Y-axis. Histograms from different genres are overlaid. Note from Figure 2e that most distributions are highly overlapping across different genres, thereby putting into question if they are indeed suitable for genre identification. But when we performed ANOVA and Kruskal-Wallis tests to figure out if the distributions were similar or not across different genres, we observed that the corresponding p-values were all < 0.001 (cf. Table 5, top two rows), thereby showing that at least a pair of mean values were significantly apart. Follow-up experiments using unpaired t-tests, U-tests, and Kolmogorov-Smirnov tests (all with Bonferroni Correction for multiple comparisons) showed that indeed almost all distributions across different genres were significantly apart from each other. Detailed results are in the supplementary material. This, we think, is an important and interesting finding, and needs to be delved deeper in future work.

Figure 3 shows histograms of the eight properties from Figure 2, but this time on a *single genre* (news articles), across different network types. This time we observed that many histograms are significantly apart from each other (see, e.g., Figures 3b, 3c, 3e, and 3f). ANOVA and Kruskal-Wallis tests corroborated this finding (cf. Table 5, bottom two rows). Detailed results, including t-tests, U-tests, and Kolmogorov-Smirnov tests are in the supplementary material.

4.3 Change of Global Network Properties with Gradual Addition of Edges

To see how global network properties change as we gradually add edges to a network, we took the whole news collection, and constructed a directed word collocation network on the whole collection, essentially considering the collection as a *super-document* (cf. Section 3). We studied how properties change as we consider top $k\%$ of edges in this super-network, with k ranging from 1 to 100 in steps of 1. The result is shown in Figure 4. Note that the number of connected components and the number of strongly connected components increase first, and then decrease. The number of vertices, size of the largest strongly connected component, and size of the largest connected component increase monotonically as we consider more and more collocation edges. For other properties, we see a lot of unpredictability and spikes (see, e.g., Figures 4d, 4e, 4g, and 4h), especially when the percentage of added edges is small. We performed Runs Test, Bartels Test, and Mann-Kendall Test to figure out if these trends are random, and the resulting p-values indicate that they are not random, and in fact positively correlated (i.e., *increasing*). Details of these tests are in the supplementary material. Note also that all figures except Figures 4a, 4k, and 4l show at least one *phase transition* (i.e., a “jump” or a “bend”).

5 Conclusion

We performed an exploratory analysis of global properties of word collocation networks across four different genres of text, and across different network types within the same genre. Our analyses reveal that cross-genre and within-genre variations are statistically significant, and incremental construction of collocation networks by gradually adding edges leads to non-random and positively correlated fluctuations in many global properties, some of them displaying single or multiple *phase transitions*. Future work consists of the inclusion of edge weights; exploration of other datasets, network properties, and network types; and applying those properties to the genre classification task.

Acknowledgments

We would like to acknowledge Dr Rada Mihalcea for her support. This work emerged from a class project in a graduate course on Network Science, given by Prof Réka Albert at Penn State. Sagnik Ray Choudhury provided us with the primary inspiration to do the hard work and write the paper. Finally, we thank the anonymous reviewers whose comments greatly improved this draft. All results, discussions, and comments contained herein are the sole responsibility of the author, and in no way associated with any of the above-mentioned people. The errors and omissions, if any, should be addressed to the author, and will be gratefully acknowledged.

References

Diego R. Amancio, Renato Fabbri, Osvaldo N. Oliveira Jr., Maria G. V. Nunes, and Luciano da Fontoura Costa. 2011. Opinion Discrimination Using Complex Network Features. In Luciano F. Costa, Alexandre Evsukoff, Giuseppe Mangioni, and Ronaldo Menezes, editors, *Complex Networks*, volume 116 of *Communications in Computer and Information Science*, pages 154–162. Springer Berlin Heidelberg.

Lucas Antiqueira, Thiago Alexandre Salgueiro Pardo, Maria das Graças Volpe Nunes, Osvaldo Novais Oliveira Jr., and Luciano da Fontoura Costa. 2006. Some issues on complex networks for author characterization. In Solange Oliveira Rezende and Antonio Carlos Roque da Silva Filho, editors, *Fourth Workshop in Information and Human Language Technology (TIL'06) in the Proceedings of International Joint Conference IBERAMIA-SBIA-SBRN*, Ribeiro Preto, Brazil, October 23-28. ICMC-USP.

Chris Biemann, Monojit Choudhury, and Animesh Mukherjee. 2009. Syntax is from Mars while Semantics from Venus! Insights from Spectral Analysis of Distributional Similarity Networks. In *ACL/IJCNLP (Short Papers)*, pages 245–248.

Chris Biemann, Stefanie Roos, and Karsten Weihe. 2012. Quantifying Semantics Using Complex Network Analysis. In *Proceedings of COLING*.

Monojit Choudhury and Animesh Mukherjee. 2009. The Structure and Dynamics of Linguistic Networks. In *Dynamics on and of Complex Networks*, pages 145–166. Springer.

Aaron Clauset, Cosma Rohilla Shalizi, and Mark E. J. Newman. 2009. Power-Law Distributions in Empirical Data. *SIAM Review*, 51(4):661–703, November.

Gabor Csardi and Tamas Nepusz. 2006. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695.

Ramon Ferrer i Cancho and Ricard V. Solé. 2001. The Small World of Human Language. *Proceedings: Biological Sciences*, 268(1482):pp. 2261–2265.

Olivier Ferret. 2002. Using collocations for topic segmentation and link detection. In *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.

Gerhard Heyer, Martin Läuter, Uwe Quasthoff, Thomas Wittig, and Christian Wolff. 2001. Learning Relations using Collocations. In *Proceedings of the IJCAI Workshop on Ontology Learning, Seattle, USA*.

Patrick Juola. 2006. Authorship Attribution. *Found. Trends Inf. Retr.*, 1(3):233–334, December.

Jinyun Ke and Yao Yao. 2008. Analysing Language Development from a Network Approach. *Journal of Quantitative Linguistics*, 15(1):70–99.

Jinyun Ke. 2007. Complex networks and human language. *CoRR*, abs/cs/0701135.

Moshe Koppel, Jonathan Schler, and Shlomo Argamon. 2009. Computational methods in authorship attribution. *J. Am. Soc. Inf. Sci. Technol.*, 60(1):9–26, January.

Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2007. Graph Evolution: Densification and Shrinking Diameters. *ACM Trans. Knowl. Discov. Data*, 1(1), March.

Wei Liang, Yuming Shi, Chi K. Tse, Jing Liu, Yanli Wang, and Xunqiang Cui. 2009. Comparison of co-occurrence networks of the Chinese and English languages. *Physica A: Statistical Mechanics and its Applications*, 388(23):4901 – 4909.

Wei Liang, YuMing Shi, Chi K. Tse, and YanLi Wang. 2012. Study on co-occurrence character networks from Chinese essays in different periods. *Science China Information Sciences*, 55(11):2417–2427.

Adolfo Paolo Masucci and Geoff J. Rodgers. 2006. Network properties of written human language. *Physical Review E*, 74(2):026102+, August.

Yutaka Matsuo, Yukio Ohsawa, and Mitsuru Ishizuka. 2001a. A Document as a Small World. In *Proceedings of the Joint JSAI 2001 Workshop on New Frontiers in Artificial Intelligence*, pages 444–448, London, UK, UK. Springer-Verlag.

Yutaka Matsuo, Yukio Ohsawa, and Mitsuru Ishizuka. 2001b. KeyWorld: Extracting Keywords from a Document as a Small World. In *Proceedings of the 4th International Conference on Discovery Science*, DS '01, pages 271–281, London, UK, UK. Springer-Verlag.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Texts. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 404–411, Barcelona, Spain, July. Association for Computational Linguistics.

Adilson E. Motter, Alessandro P. S. de Moura, Ying-Cheng Lai, and Partha Dasgupta. 2003. Topology of the conceptual network of language. *Physical Review E*, 65.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1998. The PageRank Citation Ranking: Bringing Order to the Web. In *Proceedings of the 7th International World Wide Web Conference*, pages 161–172, Brisbane, Australia.

Rishiraj Saha Roy, Niloy Ganguly, Monojit Choudhury, and Naveen Kumar Singh. 2011. Complex Network Analysis Reveals Kernel-Periphery Structure in Web Search Queries. In *Proceedings of SIGIR Workshop on Query Understanding and Representation*.

Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James Pennebaker. 2006. Effects of Age and Gender on Blogging. In *Proc. of AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs*, March.

Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *J. Am. Soc. Inf. Sci. Technol.*, 60(3):538–556, March.

J. T. Stevanak, David M. Larue, and Lincoln D. Carr. 2010. Distinguishing Fact from Fiction: Pattern Recognition in Texts Using Complex Networks. *CoRR*, abs/1007.3254.

Toby Walsh. 1999. Search in a Small World. In Thomas Dean, editor, *IJCAI*, pages 1172–1177. Morgan Kaufmann.

Stanley Wasserman and Katherine Faust. 1994. *Social Network Analysis: Methods and Applications*. Cambridge University Press, November.

Duncan J. Watts and Steven H. Strogatz. 1998. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):409–10.

Automatic Creation of Arabic Named Entity Annotated Corpus Using Wikipedia

Maha Althobaiti, Udo Kruschwitz, and Massimo Poesio
School of Computer Science and Electronic Engineering
University of Essex
Colchester, UK
{mjaltha, udo, poesio}@essex.ac.uk

Abstract

In this paper we propose a new methodology to exploit Wikipedia features and structure to automatically develop an Arabic NE annotated corpus. Each Wikipedia link is transformed into an NE type of the target article in order to produce the NE annotation. Other Wikipedia features - namely redirects, anchor texts, and inter-language links - are used to tag additional NEs, which appear without links in Wikipedia texts. Furthermore, we have developed a filtering algorithm to eliminate ambiguity when tagging candidate NEs. Herein we also introduce a mechanism based on the high coverage of Wikipedia in order to address two challenges particular to tagging NEs in Arabic text: rich morphology and the absence of capitalisation. The corpus created with our new method (*WDC*) has been used to train an NE tagger which has been tested on different domains. Judging by the results, an NE tagger trained on *WDC* can compete with those trained on manually annotated corpora.

1 Introduction

Supervised learning techniques are well known for their effectiveness to develop Named Entity Recognition (NER) taggers (Bikel et al., 1997; Sekine and others, 1998; McCallum and Li, 2003; Benajiba et al., 2008). The main disadvantage of supervised learning is that it requires a large annotated corpus. Although a substantial amount of annotated data is available for some languages, for other languages, including Arabic, more work is needed to enrich their linguistic resources. In fact, changing the domain or just expanding the set of classes always requires domain-specific experts and new annotated data, both of which cost time and effort. Therefore, current research focuses on approaches that require minimal human intervention to facilitate the process of moving the NE classifiers to new domains and to expand NE classes.

Semi-supervised and unsupervised learning approaches, along with the automatic creation of tagged corpora, are alternatives that avoid manually annotated data (Richman and Schone, 2008; Althobaiti et al., 2013). The high coverage and rich informational structure of online encyclopedias can be exploited for the automatic creation of datasets. For example, many researchers have investigated the use of Wikipedia's structure to classify Wikipedia articles and to transform links into NE annotations according to the link target type (Nothman et al., 2008; Ringland et al., 2009).

In this paper we present our approach to automatically derive a large NE annotated corpora from Arabic Wikipedia. The key to our method lies in the exploitation of Wikipedia's concepts, specifically anchor texts[1] and redirects, to handle the rich morphology in Arabic, and thereby eliminate the need to perform any deep morphological analysis. In addition, a capitalisation probability measure has been introduced and incorporated into the approach in order to replace the capitalisation feature that does not exist in the Arabic script. This capitalisation measure has been utilised in order to filter ambiguous Arabic NE phrases during annotation process.

The remainder of this paper is structured as follows: Section 2 illustrates structural information about Wikipedia. Section 3 includes background information on NER, including recent work. Section 4 summarises the proposed methodology. Sections 5, 6, and 7 describe the proposed algorithm in detail. The experimental setup and the evaluation results are reported and discussed in Section 8. Finally, the conclusion features comments regarding our future work.

[1] The terms 'anchor texts' and 'link labels' are used interchangeably in this paper.

Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 106–115,
Gothenburg, Sweden, April 26-30 2014.

2 The Structure of Wikipedia

Wikipedia is a free online encyclopedia project written collaboratively by thousands of volunteers, using MediaWiki[2]. Each article in Wikipedia is uniquely identified by its title. The title is usually the most common name for the entity explained in the article.

2.1 Types of Wikipedia Pages

2.1.1 Content Pages

Content pages (aka Wikipedia articles) contain the majority of Wikipedia's informative content. Each content page describes a single topic and has a unique title. In addition to the text describing the topic of the article, content pages may contain tables, images, links and templates.

2.1.2 Redirect Pages

A redirect page is used if there are two or more alternative names that can refer to one entity in Wikipedia. Thus, each alternative name is changed into a title whose article contains a redirect link to the actual article for that entity. For example, 'UK' is an alternative name for the 'United Kingdom', and consequently, the article with the title 'UK' is just a pointer to the article with the title 'United Kingdom'.

2.1.3 List_of Pages

Wikipedia offers several ways to group articles. One method is to group articles by lists. The items on these lists include links to articles in a particular subject area, and may include additional information about the listed items. For example, 'list of scientists' contains links to articles of scientists and also links to more specific lists of scientists.

2.2 The Structure of Wikipedia Articles

2.2.1 Categories

Every article in the Wikipedia collection should have at least one category. Categories should be on vital topics that are useful to the reader. For example, the Wikipedia article about the United Kingdom in Wikipedia is associated with a set of categories that includes 'Countries bordering the Atlantic Ocean', and 'Countries in Europe'.

2.2.2 Infobox

An infobox is a fixed-format table added to the top right-hand or left-hand corner of articles to provide a summary of some unifying parameters shared by the articles. For instance, every scientist has a name, date of birth, birthplace, nationality, and field of study.

2.3 Links

A link is a method used by Wikipedia to link pages within wiki environments. Links are enclosed in doubled square brackets. A vertical bar, the 'pipe' symbol, is used to create a link while labelling it with a different name on the current page. Look at the following two examples,

1 - [[a]] is labelled 'a' on the current page and links to taget page 'a'.

2 - [[$a|b$]] is labelled 'b' on the current page, but links to target page 'a'.

In the second example, the *anchor text* (aka *link label*) is 'a', while 'b', a *link target*, refers to the title of the target article. In the first example, the anchor text shown on the page and the title of the target article are the same.

3 Related Work

Current NE research seeks out adequate alternatives to traditional techniques such that they require minimal human intervention and solve deficiencies of traditional methods. Specific deficiencies include the limited number of NE classes resulting from the high cost of setting up corpora, and the difficulty of adapting the system to new domains.

One of these trends is distant learning, which depends on the recruitment of external knowledge to increase the performance of the classifier, or to automatically create new resources used in the learning stage.

Kazama and Torisawa (2007) exploited Wikipedia-based features to improve their NE machine learning recogniser's F-score by three percent. Their method retrieved the corresponding Wikipedia entry for each candidate word sequence in the CoNLL 2003 dataset and extracted a category label from the first sentence of the entry.

The automatic creation of training data has also been investigated using external knowledge. An et al. (2003) extracted sentences containing listed entities from the web, and produced a 1.8 million Korean word dataset. Their corpus

[2] An open source wiki package written in PHP

performed as well as manually annotated training data. Nothman et al. (2008) exploited Wikipedia to create a massive corpus of named entity annotated text. They transformed Wikipedia's links into named entity annotations by classifying the target articles into standard entity types[3]. Compared to MUC, CoNLL, and BBN corpora, their Wikipedia-derived corpora tend to perform better than other cross-corpus train/test pairs.

Nothman et al. (2013) automatically created massive, multilingual training annotations for named entity recognition by exploiting the text and internal structure of Wikipedia. They first categorised each Wikipedia article into named entity types, training and evaluating on 7,200 manually-labelled Wikipedia articles across nine languages: English, German, French, Italian, Polish, Spanish, Dutch, Portuguese, and Russian. Their cross-lingual approach achieved up to 95% accuracy. They transformed Wikipedia's links into named entity annotations by classifying the target articles into standard entity types. This technique produced reasonable annotations, but was not immediately able to compete with existing gold-standard data. They better aligned their automatic annotations to the gold standard corpus by deducing additional links and heuristically tweaking the Wikipedia corpora. Following this approach, millions of words in nine languages were annotated. Wikipedia-trained models were evaluated against CONLL shared task data and other gold-standard corpora. Their method outperformed Richman and Schone (2008) and Mika et al. (2008), and achieved scores 10% higher than models trained on newswire when tested on manually annotated Wikipedia text.

Alotaibi and Lee (2013) automatically developed two NE-annotated sets from Arabic Wikipedia. The corpora were built using the mechanism that transforms links into NE annotations, by classifying the target articles into named entity types. They used POS-tagging, morphological analysis, and linked NE phrases to detect other mentions of NEs that appear without links in text. By contrast, our method does not require POS-tagging or morphological analysis and just identifies unlinked NEs by matching phrases from an automatically constructed and filtered alternative names with identical terms in the articles texts, see Section 6. The first dataset created by Alotaibi and Lee (2013) is called *WikiFANE(whole)* and contains all sentences retrieved from the articles. The second set, which is called *WikiFANE(selective)*, is constructed by selecting only the sentences that have at least one named entity phrase.

4 Summary of the Approach

All of our experiments were conducted on the 26 March 2013 Arabic version of the Wikipedia dump[4]. A parser was created to handle the mediawiki markup and to extract structural information from the Wikipedia dump such as a list of redirect pages along with their target articles, a list of pairs containing link labels and their target articles in the form '*anchor text, target article*', and essential information for each article (e.g., title, body text, categories, and templates).

Many of Wikipedia's concepts such as links, anchor texts, redirects, and inter-language links have been exploited to transform Wikipedia into a NE annotated corpus. More details can be found in the next sections. Generally, the following steps are necessary to develop the dataset:

1. Classify Wikipedia articles into a specific set of NE types.
2. Identify matching text in the title and the first sentence of each article and label the matching phrases according to the article type.
3. Label linked phrases in the text according to the NE type of the target article.
4. Compile a list of alternative titles for articles and filter out ambiguous ones.
5. Identify matching phrases in the list and the Wikipedia text.
6. Filter sentences to prevent noisy sentences being included in the corpus.

We explain each step in turn in the following sections.

5 Classifying Wikipedia Articles into NE Categories

Categorising Wikipedia articles is the initial step in producing NE training data. Therefore, all Wikipedia articles need to be classified into a specific set of named entity types.

[3] The terms 'type', 'class' and 'category' are used interchangeably in this paper.

[4] http://dumps.wikimedia.org/arwiki/

5.1 The Dataset and Annotation

In order to develop a Wikipedia document classifier, we used a set of 4,000 manually classified Wikipedia articles that are available free online[5]. The set was manually classified using the ACE (2008) taxonomy and a new class (*Product*). Therefore, there were eight coarse-grained categories in total: *Facility*, *Geo-Political*, *Location*, *Organisation*, *Person*, *Vehicle*, *Weapon*, and *Product*. As our work adheres to the CoNLL definition, we mapped these classified Wikipedia articles into CoNLL NE types – namely person, location, organisation, miscellaneous, or other – based on the CoNLL 2003 annotation guidelines (Chinchor et al., 1999).

5.2 The Classification of Wikipedia Articles

Many researchers have already addressed the task of classifying Wikipedia articles into named entity types (Dakka and Cucerzan, 2008; Tardif et al., 2009). Alotaibi and Lee (2012) is the only study that has experimented with classifying the Arabic version of Wikipedia into NE classes. They have explored the use of Naive Bayes, Multinomial Naive Bayes, and SVM for classifying Wikipedia articles, and achieved a F-score ranging from 78% and 90% using different language-dependent and independent features.

We conducted three experiments that used a simple bag-of-words features extracted from different portions of the Wikipedia document and metadata. We summarise the portions of the document included in each experiment below:

Exp1: Experiment 1 involved tokens from the article title and the entire article body.

Exp2: Rich metadata in Wikipedia proved effective for the classification of articles (Tardif et al., 2009; Alotaibi and Lee, 2012). Therefore, in Experiment 2 we included tokens from categories, templates – specifically 'Infobox' – as well as tokens from the article title and first sentence of the document.

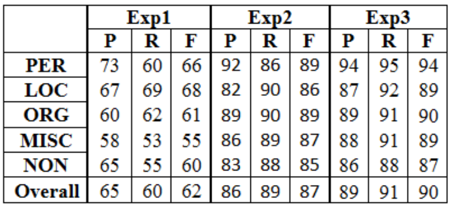
Exp3: Experiment 3 involved the same set of tokens as experiment 2 except that categories and infobox features were marked with suffixes to differentiate them from tokens extracted from the article body text. This step of distinguishing tokens based on their location in the document improved the accuracy of document's classification (Tardif et al., 2009; Alotaibi and Lee, 2012).

In order to optimise features, we implemented a filtered version of the bag-of-words article representation (e.g., removing punctuation marks and symbols) to classify the Arabic Wikipedia documents instead of using a raw dataset (Alotaibi and Lee, 2012). In addition, the same study shows the high impact of applying tokenisation[6] as opposed to the neutral effect of using stemming. We used the filtered features proposed in the study of Alotaibi and Lee (2012), which included removing punctuation marks, symbols, filtering stop words, and normalising digits. We extended the features, however, by utilising the tokenisation scheme that involves separating conjunctions, prepositions, and pronouns from each word.

The feature set has been represented using Term Frequency-Inverse Document Frequency ($TF - IDF$). This representation method is a numerical statistic that reflects how important a token is to a document.

5.3 The Results of Classifying the Wikipedia Articles

As for the learning process, our Wikipedia documents classifier was trained using Liblinear[7]. 80% of the 4,000 hand-classified Wikipedia articles were dedicated to the training stage, while 20% were specified to test the classifier. Table 1 is a comparison of the precision, recall, and F-measure of the classifiers that resulted from the three experiments. The Exp3 classifier performed better than the other classifiers. Therefore, it was selected to classify all of the Wikipedia articles. At the end of this stage, we obtained a list of pairs containing each Wikipedia article and its NE Type. We stored this list in a database in preparation for the next stage: developing the NE-tagged training corpus.



	Exp1			Exp2			Exp3		
	P	R	F	P	R	F	P	R	F
PER	73	60	66	92	86	89	94	95	94
LOC	67	69	68	82	90	86	87	92	89
ORG	60	62	61	89	90	89	89	91	90
MISC	58	53	55	86	89	87	88	91	89
NON	65	55	60	83	88	85	86	88	87
Overall	65	60	62	86	89	87	89	91	90

Table 1: The results of the three Wikipedia document classifiers.

[5] www.cs.bham.ac.uk/∼fsa081/

[6] It is also called decliticization or segmentation.

[7] www.csie.ntu.edu.tw/∼cjlin/liblinear/

6 The Annotation Process

6.1 Utilising the Titles of Articles and Link Targets

Identifying corresponding words in the article title and the entire body of text and then tagging the matching phrases with the NE-type can be a risky process, especially for terms with more than one meaning. For example, the title of the article describing the city (كان, 'Cannes')[8] can also, in Arabic, refer to the past verb (كان, 'was'). The portion of the Wikipedia article unlikely to produce errors during the matching process is the first sentence, which usually contains the definition of the term the Wikipedia article is written about (Zesch et al., 2007).

When identifying matching terms in the article title and the first sentence, we found that article titles often contain abbreviations, while the first sentence spells out entire words. This pattern makes it difficult to identify matching terms in the title and first sentence, and frequently appears in biographical Wikipedia articles. For example, one article is entitled (ابو بكر الرازي, 'Abu Bakr Al-Razi'), but the first sentence states the full name of the person: (ابو بكر محمد بن يحيى بن زكريا الرازي, 'Abu Bakr Mohammad Bin Yahia Bin Zakaria Al-Razi'). Therefore, we decided to address the problem with partial matching. In this case, the system should first identify all corresponding words in the title and the first sentence. Second, the system should annotate them and all words that fall between, provided that:

- the sequence of the words in the article title and the text are the same in order to avoid errors in tagging. For example, if the title of the article is (التايمز نهر, 'The River Thames'), but the first sentence reads (التايمز هو نهر يقع في .... , 'The Thames is a river flowing through southern England....'), then the text will not be properly tagged.
- the number of tokens located between matched tokens is less than or equal to five[9].

Figure 1 shows one example of partial matching.

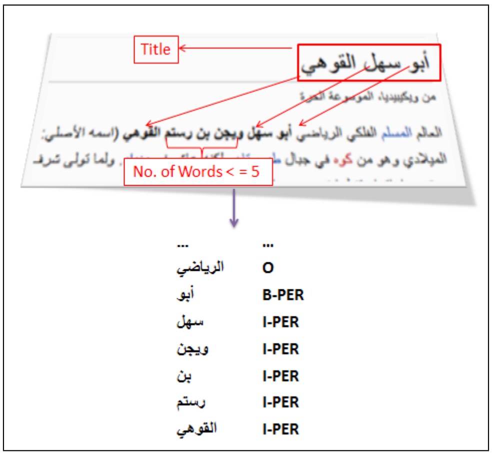

Title
أبو سهل القوهي
من ويكيبيديا، الموسوعة الحرة
العالم المسلم الفلكي الرياضي أبو سهل ويجن بن رستم القوهي (اسمه الأصلي:
الميلادي وهو من كوه في جبال ... ولما تولى شرف
No. of Words < = 5
... ...
الرياضي O
أبو B-PER
سهل I-PER
ويجن I-PER
بن I-PER
رستم I-PER
القوهي I-PER

Figure 1: Example of Partial Matching

The next step is to transform the links between Wikipedia articles into NE annotations according to the link target type. Therefore, the link ([[باراك اوباما|اوباما]]/[[Barack Obama|Obama]]) would be changed to (اوباما *PER*) (Obama *PER*), since the link target (Barack Obama) is the title of an article about person. By the end of this stage, all NE anchor texts (anchor texts referring to NE articles) on Wikipedia should be annotated based on the NE-type of the target article.

6.2 Dictionaries of Alternative Names

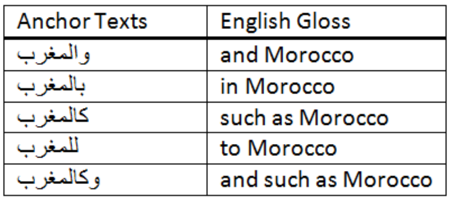
Depending only on NE anchor texts in order to derive and annotate data from Wikipedia results in a low-quality dataset, as Wikipedia contains a fair amount of NEs mentioned without links. This can be attributed to the fact that each term on Wikipedia is more likely to be linked only on its first appearance in the article (Nothman et al., 2008). These unlinked NE phrases can be found simply by identifying the matching terms in the list of linked NE phrases[10] and the text. The process is not as straightforward as it seems, however, because identifying corresponding terms may prove ineffective, especially in the case of morphologically rich language in which unlinked NE phrases are sometimes found agglutinated to prefixes and conjunctions. In order to detect unlinked and inflected forms of NEs in Wikipedia text, we extended the list of articles titles that were used in the previous step to find and match the possible NEs in the text by including NE anchor texts. Adding NE anchor texts to the list assists in finding possible morphologically inflected NEs in the text while eliminating the need for any morpho-

[8] Throughout the entire paper, Arabic words are represented as follows: (Arabic word,'English translation').

[9] An informal experiment showed that the longest proper Arabic names are 5 to 7 tokens in length.

[10] The list of anchor texts that refer to NE articles

logical analysis. Table 2 shows examples from the dictionary of NE anchor texts.



Anchor Texts	English Gloss
والمغرب	and Morocco
بالمغرب	in Morocco
كالمغرب	such as Morocco
للمغرب	to Morocco
وكالمغرب	and such as Morocco

Table 2: Examples from the dictionary of NE Anchor Texts.

Spelling variations resulting from varied transliteration of foreign named entities in some cases prevent the accurate matching and identification of some unlinked NEs, if only the list of NE anchor texts is used. For example, (انجلترا, 'England') has been written five different ways: (انجلتره, انغلترا, انغلتره, انكلترا, انكلتره). Therefore, we compiled a list of the titles of redirected pages that send the reader to articles describing NEs. We refer to these titles in this paper as *NE redirects*. We consider to the lists of NE redirects and anchor texts a list of alternative names, since they can be used as alternative names for article titles.

The list of alternative names is used to find unlinked NEs in the text by matching phrases from the list with identical terms in the articles texts. This list is essential for managing spelling and morphological variations of unlinked NEs, as well as misspelling. Consequently, the process increases the coverage of NE tags augmented within the plain texts of Wikipedia articles.

6.2.1 Filtering the Dictionaries of Alternative Names

One-word alternative names: Identifying matching phrases in the list of alternative names and the text inevitably results in a lower quality corpus due to noisy names. The noisy alternative names usually occur with meaningful named entities. For example, the article on the person (ابو عبدالله الامين, 'Abu Abdullah Alamyn') has an alternative name consisting only of his last name (الامين, 'Alameen'), which means 'custodian'. Therefore, annotating every occurrence of 'Alamyn' as *PER* would lead to incorrect tagging and ambiguity. The same applies to the city with the name (الجديده, 'Aljadydah'), which literally means 'new'. Thus, the list of alternative names should be filtered to omit one-word NE phrases that usually have a meaning and are ambiguous when taken out of context.

In order to solve this problem, we introduced a capitalisation probability measure for Arabic words, which are never capitalised. This involved finding the English gloss for each one-word alternative name and then computing its probability of being capitalised using the English Wikipedia. To find the English gloss for Arabic words, we exploited Wikipedia Arabic-to-English cross-lingual links that provided us with a reasonable number of Arabic and corresponding English terms. If the English gloss for the Arabic word could not be found using inter-language links, we resorted to an online translator. Before translating the Arabic word, a light stemmer was used to remove prefixes and conjunctions in order to get the translation of the word itself without its associated affixes. Otherwise, the Arabic word (للبلاد) would be translated as (in the country). The capitalisation probability was computed as follows:

$$Pr[EN] = \frac{f(EN)_{isCapitalised}}{f(EN)_{isCapitalised} + f(EN)_{notCapitalised}}$$

where: EN is the English gloss of the alternative name; $f(EN)_{isCapitalised}$ is the number of times the English gloss EN is capitalised in English Wikipedia; and $f(EN)_{notCapitalised}$ is the number of times the English gloss EN is not capitalised in English Wikipedia.

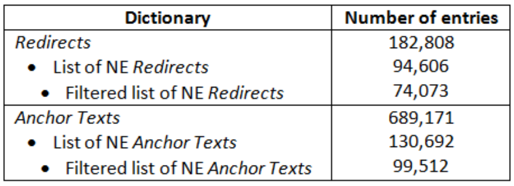
This way, we managed to build a list of Arabic words and their probabilities of being capitalised. It is evident that the meaningful one-word NEs usually achieve a low probability. By specifying a capitalisation threshold constraint, we prevented such words from being included in the list of alternative names. After a set of experiments, we decided to use the capitalisation threshold equal to 0.75.

Multi-word alternative names: Multi-word alternative names (e.g., مصطفى محمود /'MusTafae Mahmud'), احمد عادل /'Ahmad Adel') rarely cause errors in the automatic annotation process. Wikipedians, however, at times append personal and job titles to the person's name contained in the anchor text, which refers to the article about that person. Examples of such anchor texts are (حاكم دبي محمد بن راشد, 'Ruler of Dubai Muhammad bin Rashid') and (رئيس مجلس الوزراء محمد بن راشد, 'President of the Council of Ministers Muhammad bin

Rashid'). As a result, the system will mistakenly annotate words like *Dubai, Council, Ministers* as *PER*. Our solution to this problem is to omit the multi-word alternative name, if any of its words belong to the list of apposition words, which usually appear adjacent to NEs such as (رئيس, 'President'), (وزير, 'Minister'), and (حاكم, 'Ruler'). The filtering algorithm managed to exclude 22.95% of the alternative names from the original list. Algorithm 1 shows pseudo code of the filtering algorithm.

```
Algorithm 1: Filtering Alternative Names
Input: A set $L = \{l_1, l_2, \ldots, l_n\}$ of all alternative
       names of Wikipedia articles
Output: A set $RL = \{rl_1, rl_2, \ldots, rl_n\}$ of reliable
        alternative names
for $i \leftarrow 1$ to $n$ do
    $T \leftarrow$ split $l_i$ into tokens
    if ($T.size() >= 2$) then
        /* All tokens of T do not
           belong to apposition list
           */
        if (! containAppositiveWord(T)) then
            add $l_i$ to the set $RL$
    else
        $light_{stem} \leftarrow$ findLightStem($l_i$)
        $english_{gloss} \leftarrow$ translate($light_{stem}$)
        /* Compute Capitalisation
           Probability for English
           gloss                        */
        $cap_{prob} \leftarrow$ compCapProb($english_{gloss}$)
        if ($cap_{prob} > 0.75$) then
            add $l_i$ to the set $RL$
```

The dictionaries derived from Wikipedia by exploiting Wikipedia's structure and adopting the filtering algorithm is shown in Table 3.



Dictionary	Number of entries
Redirects	182,808
• List of NE *Redirects*	94,606
• Filtered list of NE *Redirects*	74,073
Anchor Texts	689,171
• List of NE *Anchor Texts*	130,692
• Filtered list of NE *Anchor Texts*	99,512

Table 3: Dictionaries derived from Wikipedia.

6.3 Post-processing

The goal of Post-processing was to address some issues that arose during the annotation process as a result of different domains, genres, and conventions of entity types. For example, nationalities and other adjectival forms of nations, religions, and ethnic groups are considered *MISC* in the CoNLL NER task in the English corpus, while the Spanish corpus consider them *NOT* named entities (Nothman et al., 2013). As far as we know, almost all Arabic NER datasets that followed the CoNLL style and guidelines in the annotation process consider nationalities *NOT* named entities. On Wikipedia all nationalities are linked to articles about the corresponding countries, which makes the annotation tool tag them as *LOC*. We decided to consider them *NOT* named entities in accordance with the CoNLL-style Arabic datasets. Therefore, in order to resolve this issue, we compiled a list of nationalities, and other adjectival forms of religion and ethnic groups, so that any anchor text matching an entry in the list was retagged as a *NOT* named entity.

The list of nationalities and apposition words used in section 6.2.1 were compiled by exploiting the 'List of' articles in Wikipedia such as *list of people by nationality*, *list of ethnic groups*, *list of adjectival forms of place names*, and *list of titles*. Some English versions of these 'List of' pages have been translated into Arabic, either because they are more comprehensive than the Arabic version, or because there is no corresponding page in Arabic.

7 Building the Corpus

After the annotation process, the last step was to incorporate sentences into the corpus. This resulted in obtaining an annotated dataset with around ten million tokens. However, in order to obtain a corpus with a large number of tags without affecting its quality, we created a dataset called Wikipedia-derived corpus (*WDC*), which included only sentences with at least three annotated named entity tokens. The *WDC* dataset contains 165,119 sentences consisting of around 6 million tokens. The annotation style of the *WDC* dataset followed the CoNLL format, where each token and its tag are placed together in the same file in the form $< token > \backslash s < tag >$. The NE boundary is specified using the *BIO* representation scheme, where *B-* indicates the beginning of the NE, *I-* refers to the continuation (Inside) of the NE, and O indicates that the word is not a NE. The *WDC* dataset is available online to the community of researchers[11]

[11] https://www.dropbox.com/sh/27afkiqvlpwyfq0/1hwWGqAcTL

8 Experimental Evaluation

To evaluate the quality of the methodology, we used *WDC* as training data to build an NER model. Then we tested the resulting classifier on datasets from different domains.

8.1 Datasets

For the evaluation purposes, we used three datasets: ANERcorp, NEWS, and TWEETS.

ANERcorp is a news-wire domain dataset built and tagged especially for the NER task by Benajiba et al. (2007). It contains around 150k tokens and is available for free. We tested our methodology on the ANERcorp test corpus because it is widely used in the literature for comparing with existing systems. The NEWS dataset is also a news-wire domain dataset collected by Darwish (2013) from the RSS feed of the Arabic version of news.google.com from October 2012. The RSS consists of the headline and the first 50 to 100 words in the news articles. This set contains approximately 15k tokens. The third test set was extracted randomly from Twitter and contains a set of 1,423 tweets authored in November 2011. It has approximately 26k tokens (Darwish, 2013).

8.2 Our Supervised Classifier

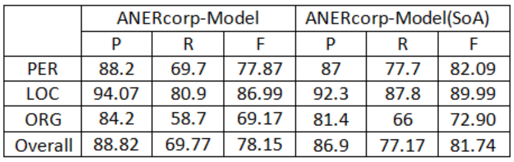
All experiments to train and build a probabilistic classifier were conducted using Conditional Random Fields (CRF)[12]. Regarding the features used in all our experiments, we selected the most successful features from Arabic NER work (Benajiba et al., 2008; Abdul-Hamid and Darwish, 2010; Darwish, 2013). These features include:

- The words immediately before and after the current word in their raw and stemmed forms.
- The first 1, 2, 3, 4 characters in a word.
- The last 1, 2, 3, 4 characters in a word.
- The appearance of the word in the gazetteer.
- The stemmed form of the word.

The gazetteer used contains around 5,000 entries and was developed by Benajiba et al. (2008). A light stemmer was used to determine the stem form of the word by using simple rules to remove conjunctions, prepositions, and definite articles (Larkey et al., 2002).

8.3 Training the Supervised Classifier on Manually-annotated Data

The supervised classifier in Section 8.2 was trained on the ANERcorp training set. We refer to the resulting model as the *ANERcorp-Model*. Table 4 shows the results of the *ANERcorp-Model* on the ANERcorp test set. The table also shows the results of the state-of-the-art supervised classifier '*ANERcorp-Model(SoA)*' developed by Darwish (2013) when trained and tested on the same datasets used for *ANERcorp-Model*.

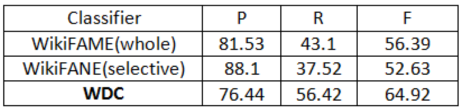


	ANERcorp-Model			ANERcorp-Model(SoA)		
	P	R	F	P	R	F
PER	88.2	69.7	77.87	87	77.7	82.09
LOC	94.07	80.9	86.99	92.3	87.8	89.99
ORG	84.2	58.7	69.17	81.4	66	72.90
Overall	88.82	69.77	78.15	86.9	77.17	81.74

Table 4: The results of Supervised Classifiers.

8.4 Results

We compared a system trained on *WDC* with the systems trained by Alotaibi and Lee (2013) on two datasets, *WikiFANE(whole)* and *WikiFANE(selective)*, which are also automatically collected from Arabic Wikipedia. The evaluation process was conducted by testing them on the ANERcorp set. The results shown in Table 5 prove that the methodology we proposed in this paper produces a dataset that outperforms the two other datasets in terms of recall and F-measure.



Classifier	P	R	F
WikiFAME(whole)	81.53	43.1	56.39
WikiFANE(selective)	88.1	37.52	52.63
WDC	76.44	56.42	64.92

Table 5: Comparison of the system trained on *WDC* dataset with the systems trained on *WikiFANE* datasets.

Table 6 compares the results of the ANERcorp-Model and the WDC-Model when testing them on datasets from different domains. Firstly, We decided to test the ANERcorp-Model and the WDC-Model on Wikipedia. Thus, a subset, containing around 14k tokens, of WDC set was allocated for testing purpose. The results in Table 6 shows that WDC classifier outperforms the F-score of the news-based classifier by around 48%.The obvious difference in the performance of the two classifiers can be attributed to the difference in annotation convention for different domains. For example, many key words in Arabic Wikipedia,

[12] http://www.chokkan.org/software/crfsuite/

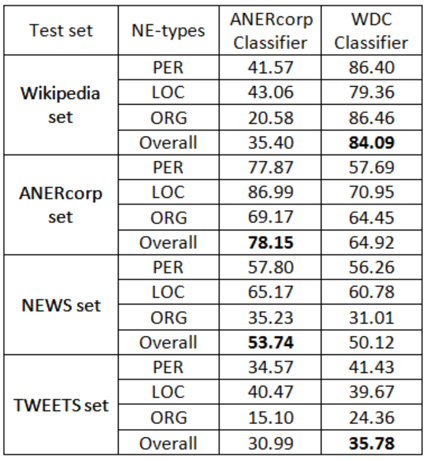
which appear in the text along with NEs (e.g., جامعة/university, مدينة/ city, شركة/company), are usually considered part of NE names. So, the phrase 'Shizuoka Prefecture' that is mentioned in some Arabic Wikipedia articles is considered an entity and linked to an article that talks about Shizuoka, making the system annotate all words in the phrase as NEs as follows: (محافظة *B-LOC* شيزوكا *I-LOC*/ Shizuoka *B-LOC* Prefecture *I-LOC*). On the other hand, in ANERcorp corpus, only the the word after the keyword (ولاية, 'Prefecture') is considered NE. In addition, although sport facilities (e.g., stadiums) are categorized in Wikipedia as *location*, some of them are not even considered entities in ANERcorp test corpus.

Secondly, the ANERcorp-Model and the WDC-Model were tested on the ANERcorp test data. The point of this comparison is to show how well the *WDC* dataset works on a news-wire domain, which is more specific than Wikipedia's open domain. The table shows that the ANERcorp-model outperforms the F-score of the WDC-Model by around 13 points. However, in addition to the fact that training and test datasets for the ANERcorp-Model are drawn from the same domain, 69% of NEs in the test data were seen in the training set (Darwish, 2013).

Thirdly, the ANERcorp-Model and the WDC-Model were tested on NEWS corpus, which is also a news-wire based dataset. The results from Table 6 reveal the quality of the *WDC* dataset on the NEWS corpus. The WDC-Model achieves relatively similar results to the ANERcorp-Model, although the latter has the advantage of being trained on a manually annotated corpus extracted from the similar domain of the NEWS test set.

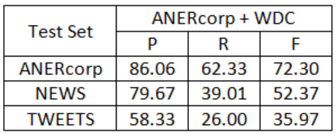
Finally, testing the ANERcorp-Model and the WDC-Model on data extracted from a social networks like Twitter proves that models trained on open-domain datasets like Wikipedia perform better on social network text than classifiers trained on domain-specific datasets, as shown in Table 6.

In order to show the effect of combining our corpus (*WDC*) with a manually annotated dataset from a different domain, we merged *WDC* with the *ANERcorp* dataset. Table 7 shows the results of a system trained on the combined corpus when testing it on three test sets. The system trained on the combined corpus achieves results that fall between the results of the systems trained on each corpus separately when testing them on the ANERcorp test set and NEWS test set. On the other hand, the results of the system trained on the combined corpus when tested on the third test set (TWEETS) show no significant improvement.



Test set	NE-types	ANERcorp Classifier	WDC Classifier
Wikipedia set	PER	41.57	86.40
	LOC	43.06	79.36
	ORG	20.58	86.46
	Overall	35.40	**84.09**
ANERcorp set	PER	77.87	57.69
	LOC	86.99	70.95
	ORG	69.17	64.45
	Overall	**78.15**	64.92
NEWS set	PER	57.80	56.26
	LOC	65.17	60.78
	ORG	35.23	31.01
	Overall	**53.74**	50.12
TWEETS set	PER	34.57	41.43
	LOC	40.47	39.67
	ORG	15.10	24.36
	Overall	30.99	**35.78**

Table 6: The F-scores of ANERcorp-Model and WDC-Model on ANERcorp, NEWS, & TWEETS datasets.



Test Set	ANERcorp + WDC		
	P	R	F
ANERcorp	86.06	62.33	72.30
NEWS	79.67	39.01	52.37
TWEETS	58.33	26.00	35.97

Table 7: The results of combining *WDC* with *ANERcorp* dataset.

9 Conclusion and Future Work

We have presented a methodology that requires minimal time and human intervention to generate an NE-annotated corpus from Wikipedia. The evaluation results showed the high quality of the developed corpus *WDC*, which contains around 6 million tokens representing different genres, as Wikipedia is considered an open domain. Furthermore, WDC outperforms other NE corpora generated automatically from Arabic Wikipedia by 8 to 12 points in terms of F-measure. Our methodology can easily be adapted to extend to new classes. Therefore, in future we intend to experiment with finer-grained NE hierarchies. In addition, we plan to carry out some domain adaptation experiments to handle the difference in annotation convention for different domains.

References

Ahmed Abdul-Hamid and Kareem Darwish. 2010. Simplified feature set for Arabic named entity recognition. In *Proceedings of the 2010 Named Entities Workshop*, pages 110–115. Association for Computational Linguistics.

Fahd Alotaibi and Mark Lee. 2012. Mapping Arabic Wikipedia into the Named Entities Taxonomy. In *COLING (Posters)*, pages 43–52.

Fahd Alotaibi and Mark Lee. 2013. Automatically Developing a Fine-grained Arabic Named Entity Corpus and Gazetteer by utilizing Wikipedia. In *IJCNLP*.

Maha Althobaiti, Udo Kruschwitz, and Massimo Poesio. 2013. A Semi-supervised Learning Approach to Arabic Named Entity Recognition. In *RANLP*, pages 32–40.

Joohui An, Seungwoo Lee, and Gary Geunbae Lee. 2003. Automatic acquisition of named entity tagged corpus from world wide web. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 2*, pages 165–168. Association for Computational Linguistics.

Yassine Benajiba, Paolo Rosso, and José Miguel Benedíruiz. 2007. Anersys: An Arabic Named Entity Recognition System based on Maximum Entropy. In *Computational Linguistics and Intelligent Text Processing*, pages 143–153. Springer.

Yassine Benajiba, Mona Diab, and Paolo Rosso. 2008. Arabic Named Entity Recognition using optimized feature sets. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 284–293. Association for Computational Linguistics.

Daniel M Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. 1997. Nymble: a high-performance learning name-finder. In *Proceedings of the fifth conference on Applied natural language processing*, pages 194–201. Association for Computational Linguistics.

Nancy Chinchor, Erica Brown, Lisa Ferro, and Patty Robinson. 1999. 1999 Named Entity Recognition Task Definition. *MITRE and SAIC*.

Wisam Dakka and Silviu Cucerzan. 2008. Augmenting Wikipedia with Named Entity Tags. In *IJCNLP*, pages 545–552.

Kareem Darwish. 2013. Named Entity Recognition using Cross-lingual Resources: Arabic as an Example. In *ACL*.

Junichi Kazama and Kentaro Torisawa. 2007. Exploiting Wikipedia as external knowledge for named entity recognition. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 698–707.

L.S. Larkey, L. Ballesteros, and M.E. Connell. 2002. Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis. In *Annual ACM Conference on Research and Development in Information Retrieval: Proceedings of the 25 th annual international ACM SIGIR conference on Research and development in information retrieval*, volume 11, pages 275–282.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 188–191. Association for Computational Linguistics.

Peter Mika, Massimiliano Ciaramita, Hugo Zaragoza, and Jordi Atserias. 2008. Learning to Tag and Tagging to Learn: A Case Study on Wikipedia. volume 23, pages 26–33.

Joel Nothman, James R Curran, and Tara Murphy. 2008. Transforming Wikipedia into Named Entity training data. In *Proceedings of the Australian Language Technology Workshop*, pages 124–132.

Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R Curran. 2013. Learning multilingual Named Entity Recognition from Wikipedia. *Artificial Intelligence*, 194:151–175.

Alexander E Richman and Patrick Schone. 2008. Mining Wiki Resources for Multilingual Named Entity Recognition. In *ACL*, pages 1–9.

Nicky Ringland, Joel Nothman, Tara Murphy, and James R Curran. 2009. Classifying articles in English and German Wikipedia. In *Australasian Language Technology Association Workshop 2009*, page 20.

Satoshi Sekine et al. 1998. NYU: Description of the Japanese NE system used for MET-2. In *Proc. of the Seventh Message Understanding Conference (MUC-7)*, volume 17.

Sam Tardif, James R. Curran, and Tara Murphy. 2009. Improved Text Categorisation for Wikipedia Named Entities. In *Proceedings of the Australasian Language Technology Association Workshop*, pages 104–108.

Torsten Zesch, Iryna Gurevych, and Max Mühlhäuser. 2007. Analyzing and accessing Wikipedia as a lexical semantic resource. pages 197–205. Tuebingen, Germany: Gunter Narr, Tübingen.

Generating artificial errors for grammatical error correction

Mariano Felice
Computer Laboratory
University of Cambridge
United Kingdom
mf501@cam.ac.uk

Zheng Yuan
Computer Laboratory
University of Cambridge
United Kingdom
zy249@cam.ac.uk

Abstract

This paper explores the generation of artificial errors for correcting grammatical mistakes made by learners of English as a second language. Artificial errors are injected into a set of error-free sentences in a probabilistic manner using statistics from a corpus. Unlike previous approaches, we use linguistic information to derive error generation probabilities and build corpora to correct several error types, including open-class errors. In addition, we also analyse the variables involved in the selection of candidate sentences. Experiments using the NUCLE corpus from the CoNLL 2013 shared task reveal that: 1) training on artificially created errors improves precision at the expense of recall and 2) different types of linguistic information are better suited for correcting different error types.

1 Introduction

Building error correction systems using machine learning techniques can require a considerable amount of annotated data which is difficult to obtain. Available error-annotated corpora are often focused on particular groups of people (e.g. non-native students), error types (e.g. spelling, syntax), genres (e.g. university essays, letters) or topics so it is not clear how representative they are or how well systems based on them will generalise. On the other hand, building new corpora is not always a viable solution since error annotation is expensive. As a result, researchers have tried to overcome these limitations either by compiling corpora automatically from the web (Mizumoto et al., 2011; Tajiri et al., 2012; Cahill et al., 2013) or using artificial corpora which are cheaper to produce and can be tailored to their needs.

Artificial error generation allows researchers to create very large error-annotated corpora with little effort and control variables such as topic and error types. Errors can be injected into candidate texts using a deterministic approach (e.g. fixed rules) or probabilities derived from manually annotated samples in order to mimic real data.

Although artificial errors have been used in previous work, we present a new approach based on linguistic information and evaluate it using the test data provided for the CoNLL 2013 shared task on grammatical error correction (Ng et al., 2013).

Our work makes the following contributions. First, we are the first to use linguistic information (such as part-of-speech (PoS) information or semantic classes) to characterise contexts of naturally occurring errors and replicate them in error-free text. Second, we apply our technique to a larger number of error types than any other previous approach, including open-class errors. The resulting datasets are used to train error correction systems aimed at learners of English as a second language (ESL). Finally, we provide a detailed description of the variables that affect artificial error generation.

2 Related work

The use of artificial data to train error correction systems has been explored by other researchers using a variety of techniques.

Izumi et al. (2003), for example, use artificial errors to target article mistakes made by Japanese learners of English. A corpus is created by replacing *a*, *an*, *the* or the zero article by a different article chosen at random in more than 7,500 correct sentences and used to train a maximum entropy model. Results show an improvement for omission errors but no change for replacement errors.

Brockett et al. (2006) describe the use of a statistical machine translation (SMT) system for correcting a set of 14 countable/uncountable nouns

Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 116–126,
Gothenburg, Sweden, April 26-30 2014. ©2014 Association for Computational Linguistics

which are often confusing for ESL learners. Their training corpus consists of a large number of sentences extracted from news articles which were deliberately modified to include typical countability errors based on evidence from a Chinese learner corpus. Their approach to artificial error injection is deterministic, using hand-coded rules to change quantifiers (*much* → *many*), generate plurals (*advice* → *advices*) or insert unnecessary determiners. Experiments show their system was generally able to beat the standard Microsoft Word 2003 grammar checker, although it produced a relatively higher rate of erroneous corrections.

SMT systems are also used by Ehsan and Faili (2013) to correct grammatical errors and context-sensitive spelling mistakes in English and Farsi. Training corpora are obtained by injecting artificial errors into well-formed treebank sentences using predefined error templates. Whenever an original sentence from the corpus matches one of these templates, a pair of correct and incorrect sentences is generated. This process is repeated multiple times if a single sentence matches more than one error template, thereby generating many pairs for the same original sentence. A comparison between the proposed systems and rule-based grammar checkers show they are complementary, with a hybrid system achieving the best performance.

2.1 Probabilistic approaches

A few researchers have explored probabilistic methods in an attempt to mimic real data more accurately. Foster and Andersen (2009), for example, describe a tool for generating artificial errors based on statistics from other corpora, such as the Cambridge Learner Corpus (CLC).[1] Their experiments show a drop in accuracy when artificial sentences are used as a replacement for real incorrect sentences, suggesting that they may not be as useful as genuine text. Their report also includes an extensive summary of previous work in the area.

Rozovskaya and Roth propose more sophisticated probabilistic methods to generate artificial errors for articles (2010a) and prepositions (2010b; 2011), also based on statistics from an ESL corpus. In particular, they compile a set of sentences from the English Wikipedia and apply the following generation methods:

General

Target words (e.g. articles) are replaced with others of the same class with probability x (varying from 0.05 to 0.18). Each new word is chosen uniformly at random.

Distribution before correction (in ESL data)

Target words in the error-free text are changed to match the distribution observed in ESL error-annotated data before any correction is made.

Distribution after correction (in ESL data)

Target words in the error-free text are changed to match the distribution observed in ESL error-annotated data after corrections are made.

Native language-specific distributions

It has been observed that second language production is affected by a learner's native language (L1) (Lee and Seneff, 2008; Leacock et al., 2010). A common example is the difficulty in using English articles appropriately by learners whose L1 has no article system, such as Russian or Japanese. Because word choice errors follow systematic patterns (i.e. they do not occur randomly), this information is extremely valuable for generating errors more accurately.

L1-specific errors can be imitated by computing word confusions in an error-annotated ESL corpora and using these distributions to change target words accordingly in error-free text. More specifically, if we estimate P(source|target) in an error-tagged corpus (i.e. the probability of an incorrect *source* word being used when the correct *target* is expected), we can generate more accurate confusion sets where each candidate has an associated probability depending on the observed word. For example, supposing that a group of learners use the preposition *to* in 10% of cases where the preposition *for* should be used (that is, P(source=*to*|target=*for*)=0.10), we can replicate this error pattern by replacing the occurrences of the preposition *for* with *to* with a probability of 0.10 in a corpus of error-free sentences. When the source and target words are the same, P(source=x|target=x) expresses the probability that a learner produces the correct/expected word.

Because errors are generally sparse (and therefore error rates are low), replicating mistakes based on observed probabilities can easily lead to

[1] http://www.cup.cam.ac.uk/gb/elt/catalogue/subject/custom/item3646603/Cambridge-International-Corpus-Cambridge-Learner-Corpus/

low recall. In order to address this issue during artificial error generation, Rozovskaya et al. (2012) propose an *inflation method* that boosts confusion probabilities in order to generate a larger proportion of artificial instances. This reformulation is shown to improve F-scores when correcting determiners and prepositions.

Experiments reveal that these approaches yield better results than assuming uniform probabilistic distributions where all errors and corrections are equally likely. In particular, classifiers trained on artificially generated data outperformed those trained on native error-free text (Rozovskaya and Roth, 2010a; Rozovskaya and Roth, 2011). However, it has also been shown that using artificially generated data as a replacement for non-native error-corrected data can lead to poorer performance (Sjöbergh and Knutsson, 2005; Foster and Andersen, 2009). This would suggest that artificial errors are more useful than native data but less useful than corrected non-native data.

Rozovskaya and Roth also control other variables in their experiments. On the one hand, they only evaluate their systems on sentences that have no spelling mistakes so as to avoid degrading performance. This is particularly important when training classifiers on features extracted with linguistic tools (such as parsers or taggers) as they could provide inaccurate results for malformed input. On the other hand, the authors work on a limited set of error types (mainly articles and prepositions) which are closed word classes and therefore have reduced confusion sets. Thus, it becomes interesting to investigate how their ideas extrapolate to open-class error types, like verb form or content word errors.

Their probabilistic error generation approach has also been used by other researchers. Imamura et al. (2012), for example, applied this method to generate artificial incorrect sentences for Japanese particle correction with an inflation factor ranging from 0.0 (no errors) to 2.0 (double error rates). Their results show that the performance of artificial corpora depends largely on the inflation rate but can achieve good results when domain adaptation is applied.

In a more exhaustive study, Cahill et al. (2013) investigate the usefulness of automatically-compiled sentences from Wikipedia revisions for correcting preposition errors. A number of classifiers are trained using error-free text, automatically-compiled annotated corpora and artificial sentences generated using error probabilities derived from Wikipedia revisions and Lang-8.[2] Their results reveal a number of interesting points, namely that artificial errors provide competitive results and perform robustly across different test sets. A learning curve analysis also shows system performance increases as more training data is used, both real and artificial.

More recently, some teams have also reported improvements by using artificial data in their submissions to the CoNLL 2013 shared task. Rozovskaya et al. (2013) apply their inflation method to train a classifier for determiner errors that achieves state-of-the-art performance while Yuan and Felice (2013) use naively-generated artificial errors within an SMT framework that places them third in terms of precision.

3 Advanced generation of artificial errors

Our work is based on the hypothesis that using carefully generated artificial errors improves the performance of error correction systems. This implies generating errors in a way that resembles available error-annotated data, using similar texts and accurate injection methods. Like other probabilistic approaches, our method assumes we have access to an error-corrected reference corpus from which we can compute error generation probabilities.

3.1 Base text selection

We analyse a set of variables that we consider important for collecting suitable texts for error injection, namely:

Topic

Replicating errors on texts about the same topic as the training/test data is more likely to produce better results than out-of-domain data, as vocabulary and word senses are more likely to be similar. In addition, similar texts are more likely to exhibit suitable contexts for error injection and consequently help the system focus on particularly useful information.

Genre

In cases where no a priori information about topic is available (for example, because the test set is

[2] `http://lang-8.com/`

unknown or the system will be used in different scenarios), knowing the genre or type of text the system will process can also be useful. Example genres include expository (descriptions, essays, reports, etc.), narrative (stories), persuasive (reviews, advertisements, etc.), procedural (instructions, recipes, experiments, etc.) and transactional texts (letters, interviews, etc.).

Style/register

As with the previous aspects, style (colloquial, academic, etc.) and register (from formal written to informal spoken) also affect production and should therefore be modelled accurately in the training data.

Text complexity/language proficiency

Candidate texts should exhibit the same reading complexity as training/test texts and be written by or targeted at learners with similar English proficiency. Otherwise, the overlap in vocabulary and grammatical structures is more likely to be small and thus hinder error injection.

Native language

Because second language production is known to be affected by a learner's L1, using candidate texts produced by groups of the same L1 as the training/test data should provide more suitable contexts for error injection. When such texts are not available, using data by speakers of other L1s that exhibit similar phenomena (e.g. no article system, agglutinative languages, etc.) might also be useful. However, finding error-free texts written in English by a specific population can be difficult, which is why most approaches resort to native English text.

In our experiments, the aforementioned variables are manually controlled although we believe many of them could be assessed automatically. For example, topics could be estimated using text similarity measures, genres could be predicted using structural information and L1s could be inferred using a native language identifier.[3]

For an analysis of other variables such as domain and error distributions, the reader should refer to Cahill et al. (2013).

3.2 Error replication

Our approach to artificial error generation is similar to the one proposed by Rozovskaya and Roth (2010a) in that we also estimate probabilities in a corpus of ESL learners which are then used to distort error-free text. However, unlike them, we refine our probabilities by imposing restrictions on the linguistic functions of the words and the contexts where they occur. Because we extend generation to open-class error types (such as verb form errors), this refinement becomes necessary to overcome disambiguation issues and lead to more accurate replication.

Our work is the first to exploit linguistic information for error generation, as described below.

Error type distributions

We compute the probability of each error type $p(t)$ occurring over the total number of relevant instances (e.g. noun phrases are relevant instances for article errors). During generation, $p(t)$ is uniformly distributed over all the possible choices for the error type (e.g. for articles, choices are *a*, *an*, *the* or the zero article). Relevant instances are detected in the base text and changed for an alternative at random using the estimated probabilities. The probability of leaving relevant instances unchanged is $1 - p(t)$.

Morphology

We believe morphological information such as person or number is particularly useful for identifying and correcting specific error types, such as articles, noun number or subject-verb agreement. Thus, we compute the conditional probability of words in specific classes for different morphological contexts (such as noun number or PoS). The following example shows confusion probabilities for singular head nouns requiring *an*:

$$\text{P(source-det=}\mathit{an}|\text{target-det=}\mathit{an}_{\text{head-noun=NN}}) = 0.942$$
$$\text{P(source-det=}\mathit{the}|\text{target-det=}\mathit{an}_{\text{head-noun=NN}}) = 0.034$$
$$\text{P(source-det=}\mathit{a}|\text{target-det=}\mathit{an}_{\text{head-noun=NN}}) = 0.015$$
$$\text{P(source-det=other}|\text{target-det=}\mathit{an}_{\text{head-noun=NN}}) = 0.005$$
$$\text{P(source-det=}\varnothing|\text{target-det=}\mathit{an}_{\text{head-noun=NN}}) = 0.004$$

PoS disambiguation

Most approaches to artificial error generation are aimed at correcting closed-class words such as articles or prepositions, which rarely occur with

[3] See the First Edition of the Shared Task on Native Language Identification (Tetreault et al., 2013) at `https://sites.google.com/site/nlisharedtask2013/`

a different part of speech in the text. However, when we consider open-class error types, we should perform PoS disambiguation since the same surface form could play different roles in a sentence. For example, consider generating artificial verb form errors for the verb *to play* after observing its distribution in an error-annotated corpus. By using PoS tags, we can easily determine if an occurrence of the word *play* is a verb or a noun and thus compute or apply the appropriate probabilities:

$P(\text{source}=\textit{play} \mid \text{target}=\textit{play}_{\text{V}}) = 0.98$
$P(\text{source}=\textit{plays} \mid \text{target}=\textit{play}_{\text{V}}) = 0.02$

$P(\text{source}=\textit{play} \mid \text{target}=\textit{play}_{\text{N}}) = 0.84$
$P(\text{source}=\textit{plays} \mid \text{target}=\textit{play}_{\text{N}}) = 0.16$

Semantic classes

We hypothesise that semantic information about concepts in the sentences can shed light on specific usage patterns that may otherwise be hidden. For example, we could refine confusion sets for prepositions according to the type of object they are applied to (a location, a recipient, an instrument, etc.):

$P(\text{prep}=\textit{in} \mid \text{noun\_class}=\text{location}) = 0.39$
$P(\text{prep}=\textit{to} \mid \text{noun\_class}=\text{location}) = 0.31$
$P(\text{prep}=\textit{at} \mid \text{noun\_class}=\text{location}) = 0.16$
$P(\text{prep}=\textit{from} \mid \text{noun\_class}=\text{location}) = 0.07$
$P(\text{prep}=\varnothing \mid \text{noun\_class}=\text{location}) = 0.05$
$P(\text{prep}=\text{other} \mid \text{noun\_class}=\text{location}) = 0.03$

By abstracting from surface forms, we can also generate faithful errors for words that have not been previously observed, e.g. we may have not seen *hospital* but we may have seen *school*, *my sister's house* or *church*.

Word senses

Polysemous words with the same PoS can exhibit different patterns of usage for each of their meanings (e.g. one meaning may co-occur with a specific preposition more often than the others). For this reason, we introduce probabilities for each word sense in an attempt to capture more accurate usage. As an example, consider a hypothetical situation in which a group of learners confuse prepositions used with the word *bank* as a financial institution but they produce the right preposition when it refers to a river bed:

$P(\text{prep}=\textit{in} \mid \text{noun}=\text{bank}_1) = 0.76$
$P(\text{prep}=\textit{at} \mid \text{noun}=\text{bank}_1) = 0.18$
$P(\text{prep}=\textit{on} \mid \text{noun}=\text{bank}_1) = 0.06$

$P(\text{prep}=\textit{on} \mid \text{noun}=\text{bank}_2) = 1.00$

Although it is rare that occurrences of the same word will refer to different meanings within a document (the so-called 'one sense per discourse' assumption (Gale et al., 1992)), this is not the case when large corpora containing different documents are used for characterising and generating errors. In such scenarios, word sense disambiguation should produce more accurate results.

Table 1 lists the actual probabilities computed from each type of information and the errors they are able to generate.

4 Experimental setup

4.1 Corpora and tools

We use the NUCLE v2.3 corpus (Dahlmeier et al., 2013) released for the CoNLL 2013 shared task on error correction, which comprises error-annotated essays written in English by students at the National University of Singapore. These essays cover topics such as environmental pollution, health care, welfare, technology, etc. All the sentences were manually annotated by human experts using a set of 27 error types, but we used the filtered version containing only the five types selected for the shared task: ArtOrDet (article or determiner), Nn (noun number), Prep (preposition), SVA (subject-verb agreements) and Vform (verb form) errors. The training set of the NUCLE corpus contains 57,151 sentences and 1,161,567 tokens while the test set comprises 1,381 sentences and 29,207 tokens. The training portion of the corpus was used to estimate the required conditional probabilities and train a few variations of our systems while the test set was reserved to evaluate performance.

Candidate native texts for error injection were extracted from the English Wikipedia, controlling the variables described Section 3.1 as follows:

Topic: We chose an initial set of 50 Wikipedia articles based on keywords in the NUCLE training data and proceeded to collect related articles by following hyperlinks in their 'See also' section. We retrieved a total of 494 articles which were later preprocessed to remove

Information	Probability	Generated error types
Error type distribution	P(*error_type*)	ArtOrDet, Nn, Prep, SVA, Vform

Morphology	P(source=*determiner*|target=*determiner*, *head_noun_tag*)
	P(source=*verb_tag*|target=*verb_tag*, *subj_head_noun_tag*)
		ArtOrDet, SVA

PoS disambiguation	P(source=*word*|target=*word*, *PoS*)	Nn, Vform

Semantic classes	P(source=*determiner*|target=*determiner*, *head_noun_class*)
	P(source=*preposition*|target=*preposition*, *head_noun_class*)
		ArtOrDet, Prep

Word senses	P(source=*preposition*|*verb_sense* + *obj_head_noun_sense*)
	P(source=*preposition*|target=*preposition*, *head_noun_sense*)
	P(source=*preposition*|target=*preposition*, *dep_adj_sense*)
	P(source=*determiner*|target=*determiner*, *head_noun_sense*)
	P(source=*verb_tag*|target=*verb_tag*, *subj_head_noun_sense*)
		ArtOrDet, Prep, SVA

Table 1: Probabilities computed for each type of linguistic information. Error codes correspond to the five error types in the CoNLL 2013 shared task: ArtOrDet (article or determiner), Nn (noun number), Prep (prepositions), SVA (subject-verb agreement) and Vform (verb form).

wikicode tags, yielding 54,945 sentences and approximately 1,123,739 tokens.

Genre: Both NUCLE and Wikipedia contain expository texts, although they are not necessarily similar.

Style/register: Written, academic and formal.

Text complexity/language proficiency: Essays in the NUCLE corpus are written by advanced university students and are therefore comparable to standard English Wikipedia articles. For less sophisticated language, the Simple English Wikipedia could be an alternative.

Native language: English Wikipedia articles are mostly written by native speakers whereas NUCLE essays are not. This is the only discordant variable.

PoS tagging was performed using RASP (Briscoe et al., 2006). Word sense disambiguation was carried out using the WordNet::SenseRelate:AllWords Perl module (Pedersen and Kolhatkar, 2009) which assigns a sense from WordNet (Miller, 1995) to each content word in a text. As for semantic information, we use WordNet classes which are readily available in NLTK (Bird et al., 2009). WordNet classes respond to a classification in lexicographers' files[4] and are defined for content words as shown in Table 2, depending on their location in the hierarchy.

Part of speech	WordNet classification
Adjective	all, pertainyms, participial
Adverb	all
Noun	act, animal, artifact, attribute, body, cognition, communication, event, feeling, food, group, location, motive, object, person, phenomenon, plant, possession, process, quantity, relation, shape, state, substance, time
Verb	body, change, cognition, communication, competition, consumption, contact, creation, emotion, motion, perception, possession, social, stative, weather

Table 2: WordNet classes for content words.

Name	Composition
ED	errors based on error type distributions
MORPH	errors based on morphology
POS	errors based on PoS disambiguation
SC	errors based on semantic classes
WSD	errors based on word senses

Table 3: Generated artificial corpora based on different types of linguistic information.

4.2 Error generation

For each type of information in Table 1, we compute the corresponding conditional probabilities using the NUCLE training set. These probabilities are then used to generate six different artificial corpora using the *inflation method* (Rozovskaya et al., 2012), as listed in Table 3.

4.3 System training

We approach the error correction task as a translation problem from incorrect into correct English. Systems are built using an SMT framework and different combinations of NUCLE and our artificial corpora, where the source side contains in-

[4] http://wordnet.princeton.edu/man/lexnames.5WN.html

	Original						**Revised**					
	C	**M**	**U**	**P**	**R**	**F_1**	**C**	**M**	**U**	**P**	**R**	**F_1**
NUCLE (baseline)	181	1462	513	0.2608	0.1102	0.1549	200	1483	495	0.2878	0.1188	0.1682
ED	53	1590	150	**0.2611**	0.0323	0.0574	62	1621	141	**0.3054**	0.0368	0.0657
MORPH	74	1569	333	0.1818	0.0450	0.0722	83	1600	324	0.2039	0.0493	0.0794
POS	42	1601	99	**0.2979**	0.0256	0.0471	42	1641	99	**0.2979**	0.0250	0.0461
SC	80	1563	543	0.1284	0.0487	0.0706	87	1596	536	0.1396	0.0517	0.0755
WSD	82	1561	305	0.2119	0.0499	0.0808	91	1592	296	0.2351	0.0541	0.0879
NUCLE+ED	173	1470	411	**0.2962**	0.1053	**0.1554**	194	1489	390	**0.3322**	0.1153	**0.1712**
NUCLE+MORPH	163	1480	427	**0.2763**	0.0992	0.1460	182	1501	408	**0.3085**	0.1081	0.1601
NUCLE+POS	164	1479	365	**0.3100**	0.0998	0.1510	182	1501	347	**0.3440**	0.1081	0.1646
NUCLE+SC	162	1481	488	0.2492	0.0986	0.1413	181	1502	469	0.2785	0.1075	0.1552
NUCLE+WSD	163	1480	413	**0.2830**	0.0992	0.1469	181	1502	395	**0.3142**	0.1075	0.1602

Table 4: Evaluation of our correction systems over the original and revised NUCLE test set using the M^2 Scorer. Columns C, M and U show the number of correct, missed and unnecessary corrections suggested by each system. Results in bold show improvements over the baseline.

correct sentences and the target side contains their corrected versions. Our setup is similar to the one described by Yuan and Felice (2013) in that we train a PoS-factored phrase-based model (Koehn, 2010) using Moses (Koehn et al., 2007), Giza++ (Och and Ney, 2003) for word alignment and the IRSTLM Toolkit (Federico et al., 2008) for language modelling. However, unlike them, we do not optimise decoding parameters but use default values instead.

We build 11 different systems in total: a baseline system using only the NUCLE training set, one system per artificial corpus and other additional systems using combinations of the NUCLE training data and our artificial corpora. Each of these systems uses a single translation model that tackles all error types at the same time.

5 Results

Each system was evaluated in terms of precision, recall and F_1 on the NUCLE test data using the M^2 Scorer (Dahlmeier and Ng, 2012), the official evaluation script for the CoNLL 2013 shared task. Table 4 shows results of evaluation on the original test set (containing only one gold standard correction per error) and a revised version (which allows for alternative corrections submitted by participating teams).

Results reveal our ED and POS corpora are able to improve precision for both test sets. It is surprising, however, that the least informed dataset (ED) is one of the best performers although this seems reasonable if we consider it is the only dataset that includes artificial instances for all error types (see Table 1). Hybrid datasets containing the NUCLE training set plus an artificial corpus also generally improve precision, except for NUCLE+SC. It could be argued that the reason for this improvement is corpus size, since our hybrid datasets are double the size of each individual set, but the small differences in precision between the ED and POS datasets and their corresponding hybrid versions seem to contradict that hypothesis. In fact, results would suggest artificial and naturally occurring errors are not interchangeable but rather complementary.

The observed improvement in precision, however, comes at the expense of recall, for which none of the systems is able to beat the baseline. This contradicts results by Rozovskaya and Roth (2010a), who show their error inflation method increases recall, although this could be due to differences in the training paradigm and data. Still, results are encouraging since precision is generally preferred over recall in error correction scenarios (Yuan and Felice, 2013).

We also evaluated performance by error type on the original (Table 5) and revised (Table 6) test data using an estimation approach similar to the one in CoNLL 2013. Results show that the performance of each dataset varies by error type, suggesting that certain types of information are better suited for specific error types. In particular, we find that on the original test set, ED achieves the highest precision for article and determiners, WSD maximises precision for prepositions and SC achieves the highest recall and F_1. When using hybrid sets, results improve overall, with the highest precision being as follows: NUCLE+POS (ArtOrDet), NUCLE+ED (Nn), NUCLE+WSD

	ArtOrDet			Nn			Prep			SVA/Vform			Other		
	P	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**C**	**M**	**U**
NUCLE (b)	0.2716	0.1551	0.1974	0.4625	0.0934	0.1555	0.1333	0.0386	0.0599	0.2604	0.1016	0.1462	0	0	34
ED	**0.2813**	0.0391	0.0687	**0.6579**	0.0631	0.1152	0.0233	0.0032	0.0056	0.0000	0.0000	—	0	0	5
MORPH	0.1862	0.1058	0.1349	—	0.0000	—	0.0000	0.0000	—	0.1429	0.0041	0.0079	0	0	7
POS	0.0000	0.0000	—	0.4405	0.0934	0.1542	0.0000	0.0000	—	0.1515	0.0203	0.0358	0	0	10
SC	0.1683	0.0739	0.1027	—	0.0000	—	0.0986	**0.0932**	**0.0959**	0.0000	0.0000	—	0	0	21
WSD	0.2219	0.1029	0.1406	0.0000	0.0000	—	**0.1905**	0.0257	0.0453	0.1875	0.0122	0.0229	0	0	8
NUCLE+ED	**0.3185**	0.1348	0.1894	**0.5465**	**0.1187**	**0.1950**	0.1304	0.0386	0.0596	**0.2658**	0.0854	0.1292	0	0	35
NUCLE+MORPH	**0.2857**	0.1507	**0.1973**	0.4590	0.0707	0.1225	**0.1719**	0.0354	0.0587	**0.2817**	0.0813	0.1262	0	0	30
NUCLE+POS	**0.3384**	0.1290	0.1868	**0.4659**	0.1035	**0.1694**	**0.1884**	**0.0418**	**0.0684**	**0.2625**	0.0854	0.1288	0	0	29
NUCLE+SC	**0.2890**	0.1290	0.1784	0.4500	0.0682	0.1184	**0.1492**	**0.0868**	**0.1098**	**0.2836**	0.0772	0.1214	0	0	34
NUCLE+WSD	**0.3003**	0.1449	0.1955	**0.4667**	0.0707	0.1228	**0.1948**	**0.0482**	**0.0773**	**0.2632**	0.0813	0.1242	0	0	30

Table 5: Error type analysis of our correction systems over the original NUCLE test set using the M^2 Scorer. Columns C, M and U show the number of correct, missed and unnecessary corrections outside the main categories suggested by each system. Results in bold show improvements over the baseline.

	ArtOrDet			Nn			Prep			SVA/Vform			Other		
	P	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**C**	**M**	**U**
NUCLE (b)	0.3519	0.2026	0.2572	0.6163	0.1302	0.2150	0.2069	0.0682	0.1026	0.4105	0.1718	0.2422	0	0	34
ED	**0.4063**	0.0579	0.1014	**0.7297**	0.0684	0.1250	0.0465	0.0077	0.0132	0.1818	0.0183	0.0332	0	0	5
MORPH	0.2270	0.1311	0.1662	—	0.0000	—	0.0000	0.0000	—	0.2857	0.0092	0.0179	0	0	7
POS	0.0000	0.0000	—	0.5465	0.1169	0.1926	0.0000	0.0000	—	**0.4242**	0.0631	0.1098	0	0	10
SC	0.2112	0.0944	0.1305	—	0.0000	—	0.1088	**0.1221**	**0.1151**	0.0000	0.0000	—	0	0	21
WSD	0.2781	0.1313	0.1784	0.0000	0.0000	—	**0.2143**	0.0347	0.0598	0.2000	0.0138	0.0259	0	0	8
NUCLE+ED	**0.4334**	0.1849	**0.2592**	**0.7000**	**0.1552**	**0.2540**	0.1685	0.0575	0.0857	**0.4744**	0.1630	**0.2426**	0	0	35
NUCLE+MORPH	**0.3791**	0.2006	**0.2624**	**0.6308**	0.1017	0.1752	**0.2295**	0.0536	0.0870	**0.4714**	0.1454	0.2222	0	0	30
NUCLE+POS	**0.4601**	0.1761	0.2547	0.6087	**0.1383**	**0.2254**	**0.2424**	0.0613	0.0979	**0.4430**	0.1549	0.2295	0	0	29
NUCLE+SC	**0.3961**	0.1773	0.2450	0.6154	0.0993	0.1709	0.1844	**0.1250**	**0.1490**	**0.4848**	0.1410	0.2184	0	0	34
NUCLE+WSD	**0.3994**	0.1933	**0.2605**	**0.6308**	0.1017	0.1752	**0.2432**	**0.0690**	**0.1075**	**0.4667**	0.1535	0.2310	0	0	30

Table 6: Error type analysis of our correction systems over the revised NUCLE test set using the M^2 Scorer. Columns C, M and U show the number of correct, missed and unnecessary corrections outside the main categories suggested by each system. Results in bold show improvements over the baseline.

(Prep) and NUCLE+SC (SVA/Vform). As expected, the use of alternative annotations in the revised test set improves results but it does not reveal any qualitative difference between datasets.

Finally, when compared to other systems in the CoNLL 2013 shared task in terms of F_1, our best systems would rank 9th on both test sets. This would suggest that using an off-the-shelf SMT system trained on a combination of real and artificial data can yield better results than other machine learning techniques (Yi et al., 2013; van den Bosch and Berck, 2013; Berend et al., 2013) or rule-based approaches (Kunchukuttan et al., 2013; Putra and Szabo, 2013; Flickinger and Yu, 2013; Sidorov et al., 2013).

6 Conclusions

This paper presents early results on the generation and use of artificial errors for grammatical error correction. Our approach uses conditional probabilities derived from an ESL error-annotated corpus to replicate errors in native error-free data. Unlike previous work, we propose using linguistic information such as PoS or sense disambiguation to refine the contexts where errors occur and thus replicate them more accurately. We use five different types of information to generate our artificial corpora, which are later evaluated in isolation as well as coupled to the original ESL training data.

General results show error distributions and PoS information produce the best results, although this varies when we analyse each error type separately. These results should allow us to generate errors more efficiently in the future by using the best approach for each error type.

We have also observed that precision improves at the expense of recall and this is more pronounced when using purely artificial sets. Finally, artificially generated errors seem to be a complement rather than an alternative to genuine data.

7 Future work

There are a number of issues we plan to address in future research, as described below.

Scaling up artificial data

The experiments presented here use a small and manually selected collection of Wikipedia articles.

However, we plan to study the performance of our systems as corpus size is increased. We are currently using a larger selection of Wikipedia articles to produce new artificial datasets ranging from 50K to 5M sentences. The resulting corpora will be used to train new error correction systems and study how precision and recall vary as more data is added during the training process, similar to Cahill et al. (2013).

Reducing differences between datasets

As shown in Table 1, we are unable to produce the same set of errors for each different type of information. This is a limitation of our conditional probabilities which encode different information in each case. In consequence, comparing overall results between datasets seems unfair as they do not target the same error types. In order to overcome this problem, we will define new probabilities so that we can generate the same types of error in all cases.

Exploring larger contexts

Our current probabilities model error contexts in a limited way, mostly by considering relations between two or three words (e.g. article+noun, verb+preposition+noun, etc.). In order to improve error injection, we will define new probabilities using larger contexts, such as P(source=verb|target=verb, subject_class, auxiliary_verbs, object_class) for verb form errors. Using more specific contexts can also be useful for correcting complex error types, such as the use of pronouns, which often requires analysing coreference chains.

Using new linguistic information

In this work we have used five types of linguistic information. However, we believe other types of information and their associated probabilities could also be useful, especially if we aim to correct more error types. Examples include spelling, grammatical relations (dependencies) and word order (syntax). Additionally, we believe the use of semantic role labels can be explored as an alternative to semantic classes, as they have proved useful for error correction (Liu et al., 2010).

Mixed error generation

In our current experiments, each artificial corpus is generated using only one type of information at a time. However, having found that certain types of information are more suitable than others for correcting specific error types (see Tables 5 and 6), we believe better artificial corpora could be created by generating instances of each error type using only the most appropriate linguistic information.

Acknowledgments

We would like to thank Prof Ted Briscoe for his valuable comments and suggestions as well as Cambridge English Language Assessment, a division of Cambridge Assessment, for supporting this research.

References

Gabor Berend, Veronika Vincze, Sina Zarrieß, and Richárd Farkas. 2013. Lfg-based features for noun number and article grammatical errors. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 62–67, Sofia, Bulgaria, August. Association for Computational Linguistics.

Steven Bird, Edward Loper, and Ewan Klein. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

Ted Briscoe, John Carroll, and Rebecca Watson. 2006. The second release of the RASP system. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, COLING-ACL '06, pages 77–80, Sydney, Australia. Association for Computational Linguistics.

Chris Brockett, William B. Dolan, and Michael Gamon. 2006. Correcting ESL Errors Using Phrasal SMT Techniques. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 249–256, Sydney, Australia, July. Association for Computational Linguistics.

Aoife Cahill, Nitin Madnani, Joel Tetreault, and Diane Napolitano. 2013. Robust systems for preposition error correction using wikipedia revisions. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 507–517, Atlanta, Georgia, June. Association for Computational Linguistics.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL 2012, pages 568 – 572, Montreal, Canada.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English. In *Proceedings of the 8th Workshop on Innovative Use of NLP for Building Educational Applications*, BEA 2013, pages 22–31, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Nava Ehsan and Heshaam Faili. 2013. Grammatical and context-sensitive error correction using a statistical machine translation framework. *Software: Practice and Experience*, 43(2):187–206.

Marcello Federico, Nicola Bertoldi, and Mauro Cettolo. 2008. IRSTLM: an open source toolkit for handling large scale language models. In *Proceedings of the 9th Annual Conference of the International Speech Communication Association*, INTERSPEECH 2008, pages 1618–1621, Brisbane, Australia, September. ISCA.

Dan Flickinger and Jiye Yu. 2013. Toward more precision in correction of grammatical errors. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 68–73, Sofia, Bulgaria, August. Association for Computational Linguistics.

Jennifer Foster and Øistein Andersen. 2009. Generrate: Generating errors for use in grammatical error detection. In *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 82–90, Boulder, Colorado, June. Association for Computational Linguistics.

William A. Gale, Kenneth W. Church, and David Yarowsky. 1992. One sense per discourse. In *Proceedings of the workshop on Speech and Natural Language*, HLT '91, pages 233–237, Harriman, New York. Association for Computational Linguistics.

Kenji Imamura, Kuniko Saito, Kugatsu Sadamitsu, and Hitoshi Nishikawa. 2012. Grammar error correction using pseudo-error sentences and domain adaptation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 388–392, Jeju Island, Korea, July. Association for Computational Linguistics.

Emi Izumi, Kiyotaka Uchimoto, Toyomi Saiga, Thepchai Supnithi, and Hitoshi Isahara. 2003. Automatic Error Detection in the Japanese Learners' English Spoken Data. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 2*, ACL '03, pages 145–148, Sapporo, Japan. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition.

Anoop Kunchukuttan, Ritesh Shah, and Pushpak Bhattacharyya. 2013. Iitb system for conll 2013 shared task: A hybrid approach to grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 82–87, Sofia, Bulgaria, August. Association for Computational Linguistics.

Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel Tetreault. 2010. *Automated Grammatical Error Detection for Language Learners*. Morgan and Claypool Publishers.

John Lee and Stephanie Seneff. 2008. An analysis of grammatical errors in non-native speech in English. In Amitava Das and Srinivas Bangalore, editors, *Proceedings of the 2008 IEEE Spoken Language Technology Workshop*, SLT 2008, pages 89–92, Goa, India, December. IEEE.

Xiaohua Liu, Bo Han, Kuan Li, Stephan Hyeonjun Stiller, and Ming Zhou. 2010. SRL-based verb selection for ESL. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1068–1076, Cambridge, MA, October. Association for Computational Linguistics.

George A. Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, November.

Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2011. Mining Revision Log of Language Learning SNS for Automated Japanese Error Correction of Second Language Learners. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 147–155, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 Shared Task on Grammatical Error Correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12, Sofia, Bulgaria, August. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Comput. Linguist.*, 29(1):19–51, March.

Ted Pedersen and Varada Kolhatkar. 2009. WordNet::SenseRelate::AllWords: a broad coverage word sense tagger that maximizes semantic relatedness. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Demonstration Session*, NAACL-Demonstrations '09, pages 17–20, Boulder, Colorado. Association for Computational Linguistics.

Desmond Darma Putra and Lili Szabo. 2013. Uds at conll 2013 shared task. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 88–95, Sofia, Bulgaria, August. Association for Computational Linguistics.

Alla Rozovskaya and Dan Roth. 2010a. Training paradigms for correcting errors in grammar and usage. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 154–162, Los Angeles, California. Association for Computational Linguistics.

Alla Rozovskaya and Dan Roth. 2010b. Generating confusion sets for context-sensitive error correction. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 961–970, Cambridge, Massachusetts. Association for Computational Linguistics.

Alla Rozovskaya and Dan Roth. 2011. Algorithm selection and model adaptation for ESL correction tasks. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 924–933, Portland, Oregon. Association for Computational Linguistics.

Alla Rozovskaya, Mark Sammons, and Dan Roth. 2012. The UI system in the HOO 2012 shared task on error correction. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 272–280, Montreal, Canada. Association for Computational Linguistics.

Alla Rozovskaya, Kai-Wei Chang, Mark Sammons, and Dan Roth. 2013. The University of Illinois System in the CoNLL-2013 Shared Task. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 13–19, Sofia, Bulgaria, August. Association for Computational Linguistics.

Grigori Sidorov, Anubhav Gupta, Martin Tozer, Dolors Catala, Angels Catena, and Sandrine Fuentes. 2013. Rule-based system for automatic grammar correction using syntactic n-grams for english language learning (l2). In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 96–101, Sofia, Bulgaria, August. Association for Computational Linguistics.

Jonas Sjöbergh and Ola Knutsson. 2005. Faking errors to avoid making errors: Very weakly supervised learning for error detection in writing. In *Proceedings of RANLP 2005*, pages 506–512, Borovets, Bulgaria, September.

Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. Tense and aspect error correction for esl learners using global context. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 198–202, Jeju Island, Korea, July. Association for Computational Linguistics.

Joel Tetreault, Daniel Blanchard, and Aoife Cahill. 2013. A Report on the First Native Language Identification Shared Task. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 48–57, Atlanta, Georgia, June. Association for Computational Linguistics.

Antal van den Bosch and Peter Berck. 2013. Memory-based grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 102–108, Sofia, Bulgaria, August. Association for Computational Linguistics.

Bong-Jun Yi, Ho-Chang Lee, and Hae-Chang Rim. 2013. Kunlp grammatical error correction system for conll-2013 shared task. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 123–127, Sofia, Bulgaria, August. Association for Computational Linguistics.

Zheng Yuan and Mariano Felice. 2013. Constrained grammatical error correction using statistical machine translation. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 52–61, Sofia, Bulgaria, August. Association for Computational Linguistics.

Author Index