Abstracts of the

Israeli Seminar on Computational Linguistics

Wednesday, 16 June 2010

Tel Aviv University

Nachum Dershowitz, Shalom Lappin, and Shuly Wintner, eds.

Israeli Seminar on Computational Linguistics 2010

Wednesday, 16 June 2010

Tel Aviv University

Rosenblatt Auditorium
Wolfson Computer & Software Engineering Building

Computational linguistics and natural language processing are active research fields in Israel today, as well as popular areas of activity in industry. The Israeli Seminar on Computational Linguistics (ISCOL) is a venue for exchanging ideas, reporting on works in progress, as well as established results, forming extramural cooperations and advancing the collaboration between academia and industry. ISCOL is a continuation of a tradition that started in 1995 with a meeting at the Technion and has continued intermittently ever since.

Conference website: http://www.cs.tau.ac.il/~nachum/iscol

Organizing Commitee:

Nachum Dershowitz (Tel Aviv Univ.)
Shalom Lappin (King's College London)
Shuly Wintner (Haifa Univ.)

Local Committee (Tel Aviv Univ.)**:**

Kfir Bar
Nachum Dershowitz
Reshef Shilon

Schedule

9:15-10:00 Hello over coffee

10:00-11:40 Session I Chair: Nachum Dershowitz

0. (5') **Amiram Yehudai**: *Welcome*

1. (25') **Jonathan Berant**, Ido Dagan, Jacob Goldberger: *Global Learning of Focused Entailment Graphs*
2. (25') Yoav Goldberg, **Michael Elhadad**: *Inspecting the Structural Biases of Dependency Parsing Algorithms*
3. (15') **Meni Adler**, Yoav Goldberg, Michael Elhadad: *Ontology-based Faceted Search Engine for Halakhic Text*
4. (15') **Raphael Cohen**, Yoav Goldberg, Michael Elhadad: *Improving Hebrew Segmentation using Non-Local Features Application to Information Extraction in the Medical Domain*
5. (15') **Nathan Grunzweig**, Yoav Goldberg, Michael Elhadad: *Linguistic Search Engine: Searching Language Phenomena on Tagged Hebrew Text*

11:40-12:10 Coffee Break

12:10-13:10 Session II Chair: Shuly Wintner

1. (15') Alon Itai, **Gai Shaked**: *How to Pronounce Hebrew Names*
2. (15') **Tal Linzen**: *Hebrew Statistical Linguistics Using a Morphologically Analyzed Blog Corpus*
3. (15') **David Gabay**, Ziv Ben-Eliahu, Michael Elhadad: *Hebrew Word Segmentation using Discourse Data*
4. (15') **Yaakov HaCohen-Kerner**, Shmuel Yishai Blitz: *Experiments with Extraction of Stopwords in Hebrew*

13:10-14:10 Lunch Break

14:10-15:45 Session III Chair: Michael Elhadad

1. (25') **Omri Abend**, Roi Reichart, Ari Rappoport: *Improved Unsupervised POS Induction through Prototype Discovery*
2. (25') **Shulamit Umansky-Pesin**, Roi Reichart, Ari Rappoport: *A Multi- Domain Web-Based Algorithm for POS Tagging of Unknown Words*
3. (15') **Roi Reichart,** Raanan Fattal, Ari Rappoport: *Improved Unsupervised POS Induction Using Intrinsic Clustering Quality, a Zipfian Constraint*
4. (15') **Kfir Bar**, Nachum Dershowitz: *Using Synonyms for Arabic-to-English Example-Based Translation*
5. (15') **Reshef Shilon**: *A Hebrew-to-Arabic Syntax-based Machine Translation System*

15:45-16:15 Coffee Break

16:15-18:00 Session IV Chair: Ido Dagan

1. (25') Ronen Feldman, Benjamin Rozenfeld, Micha Y. Breakstone **(Roy Bar-Haim)**: *SSA — A Hybrid Approach to Sentiment Analysis of Stocks*
2. (25') Dmitry Davidov, **Oren Tsur**, Ari Rappoport: *Semi-Supervised Recognition of Sarcastic Sentences in Twitter and Amazon*
3. (25') Ming-Wei Chang, James Clarke, **Dan Goldwasser**, Dan Roth: *Driving Language Interpretation from the World's Response*
4. (15') Navot Akiva, Idan Dershowitz, **Moshe Koppel**: *Exploiting Synonym Choice for Distinguishing Layers of a Document*
5. (15') **Rachel Giora**: *Irony Interpretation: Will Expecting it Make a Difference?*

Global Learning of Focused Entailment Graphs

Jonathan Berant, The Blavatnik School of Computer Science, Tel-Aviv University
Ido Dagan, the Department of Computer Science, Bar-Ilan University
Jacob Goldberger, The School of Engineering, Bar-Ilan University

The *Textual Entailment (TE)* paradigm is a generic framework for applied semantic inference. The objective of TE is to recognize whether a target meaning can be inferred from a given text. For example, a Question Answering system has to recognize that *'alcohol affects blood pressure'* is inferred from *'alcohol reduces blood pressure'* to answer the question *'What affects blood pressure?'*

TE systems require extensive knowledge of entailment patterns, often captured as *entailment rules:* rules that specify a directional inference relation between two text fragments (when the rule is bidirectional this is known as paraphrasing). An important type of entailment rule refers to *propositional templates*, i.e., propositions comprising a predicate and arguments, possibly replaced by variables. The rule required for the previous example would be *'X reduce Y* $\rightarrow$ *X affect Y'*. Because facts and knowledge are mostly expressed by propositions, such entailment rules are central to the TE task. This has led to active research on broad-scale acquisition of entailment rules for predicates.

Previous work has focused on learning each entailment rule in isolation. However, it is clear that there are interactions between rules. A prominent example is that entailment is a transitive relation, and thus the rules '$X \rightarrow Y$' and '$Y \rightarrow Z$' imply the rule '$X \rightarrow Z$'. In this paper we take advantage of these global interactions to improve entailment rule learning.

First, we describe a structure termed an *entailment graph* that models entailment relations between propositional templates. Next, we show that we can present propositions according to an entailment hierarchy derived from the graph, and suggest a novel hierarchical presentation scheme for corpus propositions referring to a target concept. As in this application each graph focuses on a single concept, we term those *focused entailment graphs*.

In the core section of the paper, we present an algorithm that uses a global approach to learn the entailment relations of focused entailment graphs. We define a global function and look for the graph that maximizes that function under a transitivity constraint. The optimization problem is formulated as an *Integer Linear Program (ILP)* and solved with an ILP solver. We show that this leads to an optimal solution with respect to the global function, and demonstrate that the algorithm outperforms methods that utilize only local information by more than 10%, as well as methods that employ a greedy optimization algorithm rather than an ILP solver.

Inspecting the Structural Biases of Dependency Parsing Algorithms

Yoav Goldberg and Michael Elhadad
Ben Gurion University, Dept of Computer Science
yoavg|elhadad@cs.bgu.ac.il

Dependency Parsing, the task of inferring a dependency structure over an input sentence, has gained research attention due in part to the CoNLL shared tasks [1, 2] in which various dependency parsing algorithms were compared on various data sets. As a result, we now have a choice of several robust, efficient and accurate parsing algorithms. These parsers achieve comparable scores, yet produce qualitatively different parses. Sagae and Lavie [4] demonstrated that a simple combination scheme of the outputs of different parsers can obtain substantially improved accuracies. Nivre and McDonald [3] explore a parser-stacking approach in which the output of one parser is fed as an input to a different parser. The stacking approach also produces more accurate parses.

While we know how to produce accurate parsers and how to blend and stack their outputs, little effort was directed toward understanding the behavior of different parsing systems in terms of structures they produce and errors they make. Question such as *which linguistic phenomena are hard for parser Y?* and *what kinds of errors are common for parser Z?*, as well as the more ambitious *which parsing approach is most suitable to parse language X?*, remain largely unanswered.

The current work aims to fill this gap, with an initial methodology to identify systematic biases in various parsing models. This methodology provides an operational definition to the notion of *structural bias* of parsers. Instead of comparing two parsing systems in terms of the errors they produce, our analysis compares the output of a parsing system with a collection of gold-parsed trees, and searches for common structures which are predicted by the parser more often than they appear in the gold-trees or vice-versa. These kinds of structures represent the bias of the parsing systems, and by analyzing them we can gain important insights into the strengths and weaknesses of the parser.

We present a boosting-based algorithm for uncovering these structural biases. We have applied our methodology to 4 parsers for English: 2 transition-based systems and 2 graph-based systems. The analysis shows that the different parsers indeed exhibit different biases. It also highlights the differences between them, and sheds light on the specific behavior of each system.

References

[1] S. Buchholz and E. Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proc. of CoNLL 2006*.

[2] J. Nivre, J. Hall, S. Kubler, R. Mcdonald, J. Nilsson, S. Riedel, and D. Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proc. of EMNLP-CoNLL 2007*.

[3] J. Nivre and R. McDonald. 2008. Integrating graph-based and transition-based dependency parsers. In *Proceedings of ACL*, pp.950-958.

[4] K. Sagae and A. Lavie. 2006. Parser combination by reparsing. In *Proceedings of HLT-NAACL*, pp.129-133.

Ontology-based Faceted Search Engine for Halachic text

Meni Adler, Yoav Goldberg , Michael Elhadad
Department of Computer Science, Ben Gurion University

Faceted search has received recent attention in the field of information retrieval. Faceted search allows users to explore a dataset by filtering available information according to a given faceted classification. Such technique for accessing a collection can be essential for users unfamiliar with the domain, of their information need or unsure about the ways to achieve their goals.
A common user may get confused, for instance, while facing the 7,094 documents retrieved by a full-text search query 'שור' on the Responsa corpus. A faceted search engine would classify these results into a set of Halachic concepts, *i.e.*, קרבנות, פדיון בכורות, קידוש החודש, נזיקין, בשר בחלב, encouraging the user to continue his exploration by selecting the topic he is interested in.
In this work, we investigate various topic modeling applications for faceted search, based on Latent Dirichlet Allocation (LDA), on a corpus of 261 SHUTIM of the medieval era. We specifically compare:

- Standard token-based modeling
- Integration of a morphological disambiguator within the LDA learning algorithm.
- Integration of a manual Halachic ontology within the LDA learning algorithm.

1. LDA over disambiguated text

Latent Dirichlet Allocation (LDA) [Blei et al. 2003] is an unsupervised topic modeling algorithm. The LDA algorithm takes as input a collection of articles, and finds latent topics characterizing these articles. Each topic is represented as a distribution over words[1]. Most of the arising topics are readily interpretable, and provide a good basis for document clustering, topic-based corpus navigation and faceted search. While LDA modeling works very well for English, Hebrew presents a challenge due to its rich morphological system which results in many different word forms representing the same concept (plural vs. singular nouns, various inflections of verbs, etc).
We propose an extension to LDA which work at the Lemma, instead of the word level. The extended algorithm takes as input a collection of documents, in which each word is annotated with one or more possible lemmas. The lemmas can be assigned either from a lexicon, or from the output of a morphological disambiguator.
We experimented with LDA-based search results clustering on the full-text of the Rambam's Yad ha-hazaka. The proof of concept system is available at: http://www.cs.bgu.ac.il/~adlerm/rambamLDA.

2. Halachic Ontology

An *Ontology* is a formal representation of a set of concepts within a domain and the relationships between these concepts. In order to determine the concept set of Halachic text, we made use of the Halachic index of המכון לחקר המשפט העברי, generalizing the subcategories of each entry.

[1] Crucially, the same word can account for different topics, depending on the other words appearing in the same context.

As usual when analyzing rich domains, assigning a concrete semantic relation between a given concept pair is complicated. At this stage, we use syntactic-based relations to obtain high agreement among ontology designers:

- Concept 2 (noun) is an object of concept 1 (gerund), *e.g.*, אבידה – שטר
- Concept 2 (noun) relates to concept 1 (noun) by a preposition, *e.g.*, אפוטרופוס – נכס
- Concept 1 (noun) is the subject of Concept 2 (gerund), *e.g.*, אפוטרופוס – מחילה
- Concept 2 (noun) is a role of concept 1 (noun), *e.g.*, אפוטרופוס – קרוב
- Projection/composition of two concepts, *e.g.*, אונס – דיני ממונות
- Concept 2 (adjective) characterizes concept 1 (noun/gerund), *e.g.*, אונס שכיח
- Concept 2 (noun) if a subtype of concept 1 (noun), *e.g.*, אלמנה - אישה

As part of the ontology design, we map each of the ontology entries to a set of documents, based on the above manual index. We currently entered a set of about 1,000 concepts and their relations. The current ontology can be found at: http://www.cs.bgu.ac.il/~adlerm/rambam

We are currently collecting a dataset of Halachic documents (a corpus of SHUTIM) manually tagged using this ontology.
The described Halachic ontology can serve various applications. In particular, it can be used for evaluation of a faceted search system on Halachic documents. The issue we specifically address, is how can this manually developed ontology be integrated with the unsupervised LDA topic modeling algorithm described above.

3. Ontology Based LDA Algorithm

While the LDA algorithm proposes an interpretable set of topics, these topics are not necessarily the same as those assigned by human annotators. We would like correlate the LDA topics with the ontology concepts. This can be done either in post-processing, mapping each learned topic to an ontology concept, or better yet by extending the LDA algorithm to make use of light supervision in the form of the concept ontology. In this LDA variant, each LDA topic will be tied to an ontology concept, and the connections in the ontology would bias the LDA topic assignments.
We are investigating various algorithmic solutions to achieve this mapping from LDA-topics to a manual ontology, relying on the relations among the topics, the ontology concepts verbal description, and the structural alignment of the graph of topics with the graph of concepts. The last constraint we use when aligning LDA topics with ontology concepts is a training set of documents manually tagged by ontology concepts and their LDA-assigned topics.

Improving Hebrew Segmentation using Non-Local Features Application to Information Extraction in the Medical Domain

Raphael Cohen, Yoav Goldberg and Michael Elhadad
Department of Computer Science, Ben-Gurion University, Beer-Sheva, Israel.

In most domain-specific texts encoded in a non-Latin alphabet, many proper names, named entities and open-class lexical items are transliterated from English. Basic NLP tasks such as POS tagging, segmentation or NER perform badly on these foreign words. Failure in segmentation is detrimental to term matching within the text for Information Extraction purposes. We developed two techniques for improving segmentation using features learned from a domain specific corpus. The first method uses word frequencies in the corpus to determine segmentation. The second method identifies transliterated words by combining unsupervised classifiers trained on the corpus and a general lexicon. Evaluation of the methods using the term matching IE task yielded improvement of 10% distinct matches with 100% precision with the first method and a 24.9% improvement with 92% precision using the second. Using both methods improved term matching by 29.7%.

Information Extraction (IE) tasks commonly occur in a specific domain (Financial, Medical, Technology, etc). Such domains contain technical words, multi-word expressions and proper names specific to the domain. These words in the specific corpora may have different features than the words in a general corpus. The differences may impede the numerous NLP tasks used for IE, among them segmentation, Part of Speech (POS) tagging, parsing, and Named Entity Recognition (NER).

The phonetic transcription of a word from a source language using a different script is called transliteration. Transliterations affect Information Retrieval in two ways. First, it takes time for a transliterated word to make it into a technical lexicon, making recognition difficult. A second problem is the variability of ways a foreign word can be rendered phonetically, leading in most cases (except for very short words) to many possible spellings of the word and, therefore, making lexicon recognition more difficult. In this paper, we demonstrate how using non-local features learned from the entire domain specific corpus can improve the extraction of a Medical Lexicon concepts from medical Q/A documents from an internet forum, a task hampered by variable spelling of the same concept and agglutination of the concept.

Agglutination of words is common in many languages. The automatic detection of word boundaries, called segmentation, is not trivial in a number of languages including Hebrew (Adler and Elhadad, 2006) and Arabic (Young-Suk et al., 2003). This task is further impeded by transliterated words.

(Adler and Elhadad, 2006) combine segmentation and morpheme tagging using an HMM method. This learning method uses a lexicon to find all the possible segmentations and choose the most likely one according to tag sequences. Unknown words, a class to which most transliterations belong, are segmented in all possible ways (there are over 150 possible prefixes and suffixes) and the most likely form is chosen using the context within the same sentence. These words account for a large fraction of the errors in this method.

Here we present a method for identifying agglutinated words based on the number of word form appearances in the entire corpus. This method is applied as a secondary segmentation mechanism for words not present in the Hebrew lexicon used for the morphological analysis and segmentation by (Adler and Elhadad, 2006). We show that the improved segmentation is helpful for the task of term extraction.

We examined the abundance of transliterations in two domains in Hebrew: medical forum and gossip news. In the medical domain in Hebrew, transliterations abound due to the frequent occurrence of

names of chemicals (*e.g.,* "*retinoic acid*") and medications without an equivalent name in Hebrew. In addition, a large number of medical doctors in Israel are either non-native speakers or trained using English textbooks. All these result in frequent usage of transliterations for words denoting anatomy ("*ureter*"), symptoms ("*atrophy*") and diseases ("*hypothyroidism*"), when there exist parallel words in Hebrew Lexicon. In the domain of gossip news, foreign names as well as slang words borrowed from English can be found frequently as well. Transliterations are common in both domains: 8.5% of the words in the medical domain are transliterated and 9% in the gossip news domain.

In Arabic, Hebrew and other Semitic languages, the syllable structure is not easily derived from the spelling and even the vowels are not clearly marked. In Modern Hebrew, vowels are only written in a minority of cases. The letters used to mark vowels ("yod", "vav" and "aleph") may also be used as consonants. Some sounds may be encoded by the same letter ("p" and "f", "sh" and "s" , "b" and "v" and "k" and "c") and some sounds do not exist in Hebrew and may be transliterated in many different ways ("th" and "j").

More than 20 suffixes and prefixes may be agglutinated on any base lemma, and sometimes up to 4 distinct affixes are combined on a single lemma. Transliterated words acquire prefixes and suffixes as well.

Most previous work concerning transliterations focused on transliteration pair acquisition, *i.e.,* recognizing that two words (source, target) are equivalent, as one is a transliteration of the other. Transliteration pair acquisition includes two sub-tasks: recognizing that a lexeme contains transliteration and finding the equivalent word in the source language (Knight and Graehl, 1998, Al-Onaizan and Knight, 2002).

The first task, recognizing a transliterated word, is language dependent. It is fairly simple in languages such as Japanese in which transliterations are written in a different script than other Japanese words and are, therefore, easily identifiable. In other languages, such as Korean, Arabic and Hebrew, deciding which word needs to be back-transliterated is more complex. (Oh and Choi, 2000) suggested a method for Korean, based on supervised naïve Bayesian learning of phonemes and their combination in transliterated words and original Korean words. This method required manual tagging of the syllables in 1,900 documents as either Korean or foreign. (Baker and Brew, 2008) reported an accuracy of 96% in Korean, with a regression model trained on automatically generated data using phonetic rules instead of a manually tagged dataset.

To recognize transliterations in Arabic, (Nwesri et al., 2006) compared a lexicon-based approach with a supervised letter N-gram learning approach, suggested by (Cavnar and Trenkle, 1994), and a method based on recognizing Arabic specific patterns. The lexicon-based approach was most successful, augmented by heuristic rules, and resulted in precision of 47.7% and recall of 57.2%.

(Goldberg and Elhadad, 2008) developed a method for transliteration recognition in Hebrew based on an N-gram letter model. The method created a training set from a pronunciation dictionary automatically, thus the method is mostly unsupervised. Before applying the n-gram classifier, agglutinated affixes were manually removed from the words. This method achieved an F-Measure of 79% when assisted by a lexicon.

In this work, we extend this approach by using larger domain specific datasets for cross domain validation. We obtained significant performance improvement by combining morphological analysis and segmentation in the process of transliteration identification instead of manual segmentation as done by (Goldberg and Elhadad, 2008). Our method produces an F-measure of 93% for the medical domain and 94% for the gossip news domain.

Using the transliteration classifier for term extraction by allowing looser matching of terms in the lexicon we obtained 18% increase in term instances recognized. Combining this method with the improved segmentation we obtained an increase of 21% in recognized term instances.

Linguistic Search Engine: searching language phenomena on tagged Hebrew Text

Nathan Grunzweig and Yoav Goldberg and Michael Elhadad
Ben Gurion University of the Negev
Department of Computer Science
POB 653 Be'er Sheva, 84105, Israel
grunzwei|yoavg|elhadad@cs.bgu.ac.il

Corpus linguistics involve searching a large corpora for linguistic evidence for various language phenomena. For lack of better alternative, corpus linguists usually rely on general search engines, such as Google, to perform their research. There are several disadvantages to this approach: (a) general purpose search engines index documents, while linguists are usually more interested in sentences, (b) general purpose search engines do not provide the kind of queries linguists are interested in, and require the user to use various query "hacks" in order to approximate the desired "real" query, (c) modern search engines perform various forms of query expansions in order to provide better user-experience, but this interferes with the linguists work, and (d) the corpora indexed by general purpose search engines is not controlled. Thus, the need for a linguistic search engine arises [2].

We present a linguists search engine for Hebrew, aimed primarily for corpus linguistics usage. The search engine indexes morphologically disambiguated sentences, and allows for queries based on several word properties, as well as linear distance between words. For example, we allow queries such as "the word צד tagged as a Noun", "two proper names followed by a definite feminine adjective", or "a past verb followed by a masculine noun, with no more than two words between them".

The search engine is based on the open source Lucene platform[1], to which we added the capability of indexing based on both words and their properties. This allows for efficient search over orthographic word forms, as well as linguistic properties such as part-of-speech, lemma, gender, tense and so forth, with various level of granularity.

The text is automatically tagged using the BGU morphological disambiguator

[1] http://lucene.apache.org/

[1]. We currently index about 75M tokens (nearly 8M sentences) from various genres including blogs, news, knesset proceedings and medical articles. Queries return almost instantly. Adding additional datasets is trivial. We provide a web interface, which will soon be available at `http://www.cs.bgu.ac.il/~nlpproj/corpus-search`.

References

[1] Meni Adler. *Hebrew Morphological Disambiguation: An Unsupervised Stochastic Word-based Approach*. PhD thesis, Ben-Gurion University of the Negev, Beer-Sheva, Israel, 2007.

[2] Adam Kilgariff. Linguistic search engine. In *Proc. of Corpus Linguistics*, 2003.

How to Pronounce Hebrew Names

Alon Itai and Gai Shaked
Computer Science Department
Technion, Haifa, Israel

This paper addresses the problem of determining the correct pronunciation of People's names written in Hebrew, by extracting clues from the way the same name is written in other languages, and using a corpus of known names pronunciation to guess the correct pronunciation of a given name.

Names differs from other words in a language because they do not follow the language's fixed set of rules. Names are not necessarily composed of the morphemes of the language and its inflections. Names may have different origins, while some names are legitimate words in the language others are distorted words, originating in different language etc. Enumerating the most common names does not solve the problem because of the long tail effect – in each population there is a set of common and widespread names, most of which occur infrequently.

Two features of modern Hebrew makes the problem of determining the correct pronunciation of names written in Hebrew especially interesting. First, the consonants are written in the form of letters while the vowels written as diatritics (dots–niqqud ניקוד) . The common modern Hebrew script omits the niqqud so a word is actually a sequence of consonants and one must be familiar with the word and the context in order to add the right vowels. Second, a large portion of the population in Israel are immigrants, or descendants of immigrants. This fact creates a great variety of names, from large variety of cultures and languages.

In addition to the lack of vowels, five consonants (Bet, Kaf, Pe, Cadi and Shin) have two realizations. Also the letters Vav and Yod may either realize a vowel or a consonant.

In this paper we will regard pronunciation as *dotting* – adding the niqqud signs, or dots, to resolve the above ambiguities. Namely, we will add letters to indicate A, E, I, O or U vowels, and for each of the letters - Bet (ב), Khaf (כ), Pe (פ) and Shin (ש) we will denote which of the two possible consonants it represents.

We suggest two algorithms:

- **The Agglutinating Algorithm** Given a training set of dotted names T, and a test name w, try to match the prefix and suffix of w with prefixes and suffixes of names of T. Dot the prefix w in compliance with the best prefix match, and dot the suffix similarly.
- **The Transliteration Algorithm** Using open sources, such as Wikipedia, we match Hebrew names to their English equivalent. We use the English transcription to disambiguate the ambiguous consonants and insert the vowels.

In order to test our algorithms we took 1000 random records from the Israeli phone book, the names in the records were manually dotted and divided into two test corpora – one for first names

Training set	Records	Transliteration algorithm First Names	Transliteration algorithm Last Names	Combined algorithm First Names	Combined algorithm Last Names
Names from Wikipedia	5,358[1]	75.9%	38.8%	86.1%	56.9%
Technion students	75,997[2]	62.2%	62.8%	76.0%	84.0%

Table 1: The success rate of both algorithms when using bilingual lists of names.

and one for surnames. There is a total of 1338 first names (476 unique), and 1002 surnames (770 unique). We used three sources of dotted names to test the performance of the algorithms:

- Another set of random names from the phonebook, manually dotted and divided into first and last names.
- A list of article names from Wikipedia, all the articles in the category *people* in the Hebrew Wikipedia which have an equivalent article in English. This list was dotted using the English-Hebrew algorithm described above.
- A list of last names of the Technion students, this list contains both Hebrew and English last names of the students in the Technion–Israel Institute of Technology, throughout the years. This list was also dotted by the English-Hebrew algorithm.

The success rate of the Agglutinating algorithm when applied to first names was 83.8%; for surnames 54.2%. However when learning to pronounce surnames while training on first names we achieved a success rate of 62.2%.

Table 1 summarizes the results of both the algorithms, applied with the lists of English and Hebrew names. First we present the results obtained using the second algorithm alone. We also tested a combined algorithm: we used the dotted names generated by the transliteration algorithm as a training set for the agglutinating algorithm.

Finally, we have used the combined algorithm on all of the source lists together, and obtained a success rate of 93.4% for first names, and 87.1% for surnames. First names are clearly easier to dot, they tend to be shorter and they are less diverse than surnames. The results of the algorithm improve (logarithmicly) as the size of the training data (lists of dotted names) increases.

One may further improve the results by using Machine Learning techniques to determine which of the dottings to choose when addressed with conflicting overlap data.

Finally, we have not discussed the problem of stress: determining which syllable is stressed. First there is a discrepancy between the location of the stress in normative and colloquial Hebrew (IlAn vs. Ilan, ShlomO vs. ShlOmo). Native Hebrew speakers agree on the stress of most names which are derived from other word by adding a suffix (ShmuEli, ShkEdi). They even agree on the stress of names with non Hebrew suffixes (e.g. ShmuelOvich, RabinOvich). We did not apply our methods since we were not able to obtain a database of names that includes the stress. We hope to obtain such a database and test our methods.

Hebrew Statistical Linguistics Using a Morphologically Analyzed Blog Corpus

Tal Linzen, Tel Aviv University

This talk presents the Hebrew Blog Corpus and briefly outlines two linguistic research projects based on this corpus.

The American structuralist school of linguistics, most notably identified with Leonard Bloomfield and epitomized in his 1933 book *Language*, saw the goal of linguistics as describing in a concise way corpora of naturally occurring speech. The Chomskyan revolution challenged the primacy of corpora in linguistics; from the 1960s on, much of the argumentation in linguistics has been based on artificial sentences and subjective judgments. However, changing scientific fashions, and the proliferation of large electronic corpora, have led in recent years to a renewed interest in corpus work in linguistics.

The largest corpus in existence is the one stored on Google's servers. The search facilities Google provides are, however, limited to simple strings of words. In a practically isolating language such as English, this limitation often amounts to little more than a minor inconvenience; in a morphologically rich language such as Hebrew it can turn into an insurmountable obstacle. Google search results suffer from two other drawbacks which make them unsuitable for quantitative corpus research: they are not reproducible, nor quantifiable in a reliable way (the counts are estimates, and often behave in a rather bizarre fashion). In addition, the data Google presents to the user lack linguistic annotation and speaker metadata.

Hence the importance for linguistic research of using stable and publicly available corpora. Most of the Hebrew corpora currently available consist of texts in a formal or written register (parliament session protocols, newspaper articles). In addition, there are two relatively small informal language corpora, containing less than 3 million tokens combined. The Blog Corpus fills the need for a large, non-copy-edited corpus. It consists of 150 million tokens – 3 times all the existing corpora combined – made up of posts published at the Israblog blogging platform. A useful property of blogging sites is that they clearly indicate for each segment of text the identity of its writer, which facilitates the study of individual variation.

Limiting data acquisition to one site greatly simplifies "scraping" (HTML cleanup). This specific blogging site was chosen because of its size, and because of its handy "random blog" feature, which makes it easier to automatically discover users – in general not a trivial task. Another advantage of the site is that most of its users choose to specify their age and gender. The majority of the users are young, aged 16 to 22, but some users report ages as high as 80. This information can be used to investigate the effect of age and gender on language use.

The Blog Corpus was not designed to be balanced with respect to genres, socio-economic status and so on. There are some biases: older users are underrepresented, women are overrepresented. Still, the selection is probably more varied than the in other Hebrew corpora. The fact that the texts are not copy-edited is both a blessing (more spontaneous language) and a curse (for example, spelling mistakes, though those can be interesting in their own right).

The corpus was morphologically analyzed using Meni Adler's disambiguator (and with Yoav Goldberg's kind help). To the best of my knowledge there is no syntactic parser for Hebrew. Which is why

the texts have not be parsed.

I used this corpus for two projects so far. The first is a corpus study comparing the Hebrew Possessive Dative construction, as in (1), to the ordinary possessive construction, as in (2):

(1) šavarti le-šaul et ha-kos.
I.broke to-Shaul ACC the-glass
'I broke Shaul's glass.' *(contested gloss)*

(2) šavarti et ha-kos šel šaul.
I.broke ACC the-glass of Shaul
'I broke Shaul's glass.'

Based on data from similar constructions in other European languages, I predicted that the Possessive Dative will be used more often when the possessed object is a body part. To test this prediction I needed to search for sentences of the form (1); since the Hebrew preposition *le* 'to' is fused with its complement in Hebrew orthography, this would not have been possible without a morphologically analyzed corpus.

In the next stage of the project I conjectured that this preference for body parts is becoming less and less pronounced with time. The time of writing is more or less identical for all texts, so it is of little help; instead I used the ages reported by the users, under the assumption that texts produced by older speakers reflect previous stages of the language. Analysis using a mixed-effect logistic model supported the conjecture.

The corpus was also used to prepare materials for a neurolinguistic experiment (in collaboration with Einat Shetreet and Naama Friedmann). Verbs can often appear in more than one *frame*, or syntactic context; for instance, *want* can either take an infinitive, as in *I want to sleep*, or a noun complement, as in *I want an icecream*. Our experiment compares three classes of Hebrew verbs: verbs that can only appear in a single frame; multiple-frame verbs which nevertheless show a clear frequency bias towards a single frame; and multiple-frame verbs that are not biased towards any one frame. Following the findings of other experiments, we expect to find three different patterns of brain activity, one for each class of verbs, in a specific brain area associated with language.

To distinguish between the two classes of multiple-frame verbs, we needed to calculate for each verb the frequency of each frame in a sample of the corpus. The morphologically analyzed corpus enabled us to do much of the work automatically, namely to search for all the forms of a given verb at once, and to identify frames with an orthographically fused preposition. However, given the lack of a syntactic parser, it was very hard to accurately classify several types of sentences, such as cases of non-canonical word order, as in (3):

(3) al ma racita ledaber?
about what you.wanted to.talk
'What did you want to talk about?'

The results therefore had to undergo substantial manual revision.

Finally, in the last two years the corpus has been used in other projects: to obtain word frequency norms for psycholinguistic experiments (in Naama Friedmann's lab), in Hillel Taub-Tabib's corpus research on subject-verb inversion, in Nurit Melnik's research, and to improve parsers in Ben Gurion University.

Hebrew Word Segmentation using Discourse Data

David Gabay
Pursway.com

Ziv Ben-Eliau
ze-timeline.net

Michael Elhadad
Ben-Gurion University of the Negev
Department of Computer Science

Word segmentation and part of speech tagging in Hebrew, like similar natural language tasks, are usually done using local information, which does not go beyond the context of a single sentence [1][2]. In some cases, statistics gathered from the entire corpus are also used. In real life, we rarely encounter sentences floating without context, that could be a paragraph, a document (*e.g.,* a news story), an easily defined hierarchical class of documents (*e.g.,* a section in a newspaper, or the news stories from a given day, week or month). The probability that a given word segmentation is correct within a sentence depends heavily on those intermediate contexts. We will report on ongoing work on ways to extract information from contexts between sentence and corpus level, in particular document level, in order to resolve segmentation ambiguities in Hebrew.

The idea of augmenting ambiguity resolution using discourse data is not new. It arises naturally from the assumption that well-written text is cohesive. At the semantic level, ambiguous words tend to preserve a single meaning in all occurrences in the same article ('one sense per discourse'[3]), a tendency that was found useful for word sense disambiguation in English. A similar assumption ('one tokenization per source') was used for Chinese word tokenization [4] and Chinese and Japanese part-of-speech tagging for unknown words [5].

Hebrew word segmentation (finding the correct prefix, if it exists, of a Hebrew word in its context) is a major cause for ambiguity in Hebrew: 50% of all tokens in a news corpus can be segmented in more than one way; out-of-vocabulary tokens with different possible segmentations are particularly troublesome to existing disambiguation tools, which rely on local and global information only.

The assumption of 'one segmentation per document' was verified on two corpora, "Haaretz" news corpus a corpus of Hebrew Wikipedia article, partially tagged for segmentation[6]. The Haaretz corpus [1]is manually tagged and we verified that 9,774 of all word types appear more than once in the same article; out of which, only 59 word types have two different segmentations in the same article. This is only one of many dependencies between correct segmentations of words within a document. More generally we observe that the probability of a word of the form w=pv, where p is a possible prefix, to be segmented as p+v, depends on the number of occurrences of v and w in the same document, as well as on the overall frequency of the prefix p in the corpus.

Such dependencies may be learned from a training corpus, or developed by self-training, and used to resolve segmentation disambiguates without any language knowledge except a list of possible prefixes. Essentially, our method[2] works as follows: for each possible segmentation pattern, we count the number of occurrences of the expected lemma in the current context level. If the counts give strong indication, the segmentation is disambiguated; else we try a higher level

[1]Available at the Mila knowledge center for processing Hebrew: http://www.mila.cs.technion.ac.il/
[2]Code is available at: http://sourceforge.net/projects/duck/

context. In initial experiments, we achieved 92% correct segmentations on Wikipedia articles without any training data. Our system turns to higher context information in those cases where the segmentation cannot be determined by the document context. In the Wikipedia case, the article category served as its immediate higher context.

We also study methods of combining high context information with disambiguation systems that rely on sentence-level information, such as [1].

References

[1] Adler, M., Hebrew Morphological Disambiguation: An Unsupervised Stochastic Word-based Approach, Ph.D. thesis, 2007

[2] Bar-Haim R., Sima'an K., and Winter Y., Choosing an optimal architecture for segmentation and POS-tagging of Modern Hebrew, ACL 2005

[3] Yarowsky, D., One sense per collocation. In Proceedings of the ARPA Human Language Technology Workshop, 1993

[4] Guo, j., One Tokenization per Source, ACL 1998

[5] Nakagawa, T. and Matsumoto, Y., Guessing parts-of-speech of unknown words using global information, ACL 2006

[6] Gabay, D. and Ben-Eliau, Z., and Elhadad, M., Using Wikipedia Links to Construct Word Segmentation Corpora, WIKIAI-08 Workshop, AAAI-2008

Experiments with Extraction of Stopwords in Hebrew

Yaakov HaCohen-Kerner, Shmuel Yishai Blitz

Department of Computer Science, Jerusalem College of Technology
21 Havaad Haleumi St., P.O.B. 16031, 91160 Jerusalem, Israel
kerner@jct.ac.il, lightbulbil@gmail.com

Stopwords are very common and are widely used in many languages. These words are regarded as meaningless in terms of information retrieval. Various stopword lists have been constructed for English and a few other languages. However, to the best of our knowledge, no stopword list has been constructed for Hebrew. In this research, we present an application of three baseline methods (TFN, TFDN, IDFN) that attempt to extract stopwords for a data set containing Israeli daily news.

The examined dataset includes texts in Hebrew that were published in Arutz 7 – Israel National News (http://www.inn.co.il). All documents belong to the same domain: Free Daily Israel Reports in Hebrew from the year of 2008. The entire dataset includes 13,342 news documents. They include 3,463,871 tokens where 171,814 of them are unique. Each document includes in average 259.6 tokens, while 191.5 are unique.

In contrast to quite high overlapping rate (above 80%) between top 100 English[1] and Chinese stopwords, there is about medium overlapping rate (60%) between top occurring Hebrew and English words.

Only two words are included in the top 10 words in the English and Hebrew lists: 'of' and 'he'. Four additional overlapped words are contained in the 20 top word lists: 'not', 'on', 'that', and 'it'. The word 'the' that is placed on the first place in the English list appears only at the 87th place in the Hebrew list.

These findings might be due to: (1) The Hebrew corpus analyzed in this research which is a news corpus is not general as the English corpus (many top occurring Hebrew words are related to Israel and its politics, e.g., Israel, security, IDF, government, prime, minister, parliament) and (2) The Hebrew morphology is one of the sources for major differences between the Hebrew and English stopword results. For instance, many English articles, prepositions, and conjunctions (e.g., 'a', 'an', 'the', 'in', 'and') are usually not presented as single words in Hebrew, but rather as prefixes of the words that come immediately after.

The Zipf's law failed to describe the distribution of the top occurring words in the tested data set. A possible explanation is that the examined documents are not uniformly distributed across the categories. This is probably the case with our data set, which contains Israeli daily news.

Another important finding is the fact that TFN presents the smoothest curve among the three baseline methods. Indeed, this fact is quite trivial since the graph deals with the frequencies of top occurring words according to their place and TFN is the only method that actually expresses this relation.

Other experiments identify important topics as a function of time (months annd weeks) using content words that appear in the top occurring words. In similar to previous research, the *IDFN* method achieved the best improvement in the precision values after omitting its unique stopwords from the tested web-queries.

[1] based on the Brown corpus (Francis, 1982).

Improved Unsupervised POS Induction through Prototype Discovery

Omri Abend[1*] Roi Reichart[2] Ari Rappoport[1]

[1]Institute of Computer Science, [2]ICNC
Hebrew University of Jerusalem
{omria01|roiri|arir}@cs.huji.ac.il

Part-of-speech (POS) tagging is a fundamental NLP task, used by a wide variety of applications. However, there is no single standard POS tagging scheme, even for English. Schemes vary significantly across corpora and even more so across languages, creating difficulties in using POS tags across domains and for multi-lingual systems (Jiang et al., 2009). Automatic induction of POS tags from plain text can greatly alleviate this problem, as well as eliminate the efforts incurred by manual annotations. It is also a problem of great theoretical interest. Consequently, POS induction is a vibrant research area.

In this paper we present an algorithm based on the theory of prototypes (Taylor, 2003), which posits that some members in cognitive categories are more central than others. These practically define the category, while the membership of other elements is based on their association with the central members. Our algorithm first clusters words based on a fine morphological representation. It then clusters the most frequent words, defining *landmark* clusters which constitute the cores of the categories. Finally, it maps the rest of the words to these categories. The last two stages utilize a distributional representation that has been shown to be effective for unsupervised parsing (Seginer, 2007).

We use a morphological representation in which each word is represented by its morphological signature (Goldsmith, 2001) and its specific inclination. This information is obtained by the *Morferssor* unsupervised segmentation model (Creutz and Lagus, 2005).

We evaluated the algorithm in both English and German, using four different mapping-based and information theoretic clustering evaluation measures. The results obtained are generally better than all existing POS induction algorithms.

This work will be presented in the annual meeting of the ACL.

Bibliography:

1. Mathias Creutz and Krista Lagus, 2005. *Inducing the Morphological Lexicon of a Natural Language from Unannotated Text.* AKRR '05.

2. John Goldsmith, 2001. *Unsupervised Learning of the Morphology of a Natural Language.* Computational Linguistics, 27(2):153–198.

* Omri Abend is grateful to the Azrieli Foundation for the award of an Azrieli Fellowship.

3. Wenbin Jiang, Liang Huang and Qun Liu, 2009. *Automatic Adaptation of Annotation Standards: Chinese Word Segmentation and POS Tagging – A Case Study.* ACL '09.

4. Yoav Seginer, 2007. *Fast Unsupervised Incremental Parsing.* ACL '07.

5. John R. Taylor, 2003. *Linguistic Categorization: Prototypes in Linguistic Theory, Third Edition.* Oxford University Press.

A Multi-Domain Web-Based Algorithm for POS Tagging of Unknown Words

Shulamit Umansky-Pesin[1] Roi Reichart[2] Ari Rappoport[1]

[1]Institute of Computer Science , [2]ICNC
Hebrew University of Jerusalem
{pesin|roiri|arir}@cs.huji.ac.il

Part-of-speech (POS) tagging is a fundamental NLP task that has attracted many researchers in the last decades. While supervised POS taggers have achieved high accuracy (e.g., (Toutanova et al., 2003) report a 97.24% accuracy in the WSJ Penn Treebank), tagger performance on words appearing a small number of times in their training corpus (*unknown words*) is substantially lower. This effect is especially pronounced in the *domain adaptation* scenario, where the training and test corpora are from different domains. For example, when training the MXPOST POS tagger (Ratnaparkhi, 1996) on sections 2-21 of the WSJ Penn Treebank it achieves 97.04% overall accuracy when tested on WSJ section 24, and 88.81% overall accuracy when tested on the BNC corpus, which contains texts from various genres. For unknown words (test corpus words appearing 8 times or less in the training corpus), accuracy drops to 89.45% and 70.25% respectively.

In this paper we propose an unknown word POS tagging algorithm based on web queries. When a new sentence s containing an unknown word u is to be tagged by a trained POS tagger, our algorithm collects from the web contexts that are partially similar to the context of u in s. The collected contexts are used to compute new tag assignment probabilities for u.

To the best of our knowledge this is the first web-query based algorithm for POS tagging or for any syntactic NLP task.

Our algorithm is particularly suitable for *multi-domain* tagging, since it requires no information about the domain from which the sentence to be tagged is drawn. It does not need domain specific corpora or external dictionaries, and it requires no preprocessing step. The information required for tagging an unknown word is very quickly collected from the web.

This behavior is unlike previous works for the task (e.g (Blitzer et al., 2006)), which require a time consuming preprocessing step and a corpus collected from the target domain. When the target domain is heterogeneous (as is the web itself), a corpus representing it is very hard to assemble. To the best of our knowledge, ours is the first paper to provide such an *on-the-fly* unknown word tagging algorithm.

To demonstrate the power of our algorithm as a fast multi-domain learner, we experiment in three languages (English, German and Chinese) and several domains. We implemented the MXPOST tagger and integrated it with our algorithm. We show error reduction in unknown word tagging of up to 15.63% (English), 18.09% (German) and 13.57% (Chinese) over MXPOST.

The run time overhead is less than 0.5 seconds per an unknown word in the English and German experiments, and less than a second per unknown word in the Chinese experiments.

References

Blitzer, John, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia.

Ratnaparkhi, Adwait. 1996. A maximum entropy model for part-of-speech tagging. In *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142.

Toutanova, Kristina, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 173–180, Morristown, NJ, USA. Association for Computational Linguistics.

Improved Unsupervised POS Induction Using Intrinsic Clustering Quality and a Zipfian Constraint

Roi Reichart (ICNC, Hebrew University, roiri@cs.huji.ac.il)
Raanan Fattal (Institute of Computer Science, Hebrew University, raananf@cs.huji.ac.il)
Ari Rappoport(Institute of Computer Science, Hebrew University, arir@cs.huji.ac.il)

Unsupervised part-of-speech (POS) induction is of major theoretical and practical importance. It counters the arbitrary nature of manually designed tag sets, and avoids manual corpus annotation costs. The task enjoys considerable current interest in the research community.

Most unsupervised POS tagging algorithms apply an optimization procedure to a non-convex function, and tend to converge to local maxima that strongly depend on the algorithm's (usually random) initialization. The quality of the taggings produced by different initializations varies substantially. Figure 1 demonstrates this phenomenon for a leading POS induction algorithm [1]. The absolute variability of the induced tagging quality is 10-15%, which is around 20% of the mean. Strong variability has also been reported by other authors.

The common practice in the literature is to report mean results over several random initializations of the algorithm (e.g. [1, 2, 3, 4]). This means that applications using the induced tagging are not guaranteed to use a tagging of the reported quality.

In this paper we address this issue using an unsupervised test for intrinsic clustering quality. We present a quality-based algorithmic family Q. Each of its concrete member algorithms Q(B) runs a base tagger B with different random initializations, and selects the best tagging according the quality test. If the test is highly positively correlated with external tagging quality measures (e.g., those based on gold standard tagging), Q(B) will produce better results than B with high probability.

We experiment with two base taggers, Clark's original tagger (CT) and *Zipf Constrained Clark (ZCC)*. ZCC is a novel algorithm of interest in its own right, which is especially suitable as a base tagger in the family Q. ZCC is a modification of Clark's algorithm in which the distribution of the number of word types in a cluster (*cluster type size*) is constrained to be Zipfian. This property holds for natural languages, hence we can expect a higher correlation between ZCC and an accepted unsupervised quality measure, perplexity.

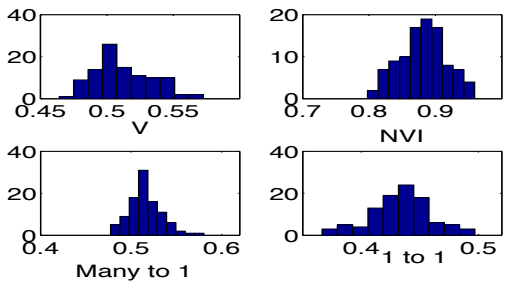

Figure 1: Distribution of the quality of the taggings produced in 100 runs of the Clark POS induction algorithm (with different random initializations) for sections 2-21 of the WSJ corpus. All graphs are 10-bin histograms presenting the number of runs (y-axis) with the corresponding quality (x-axis). Quality is evaluated with 4 clustering evaluation measures: V, NVI, greedy m-1 mapping and greedy 1-1 mapping. The quality of the induced tagging varies considerably.

We show that for both base taggers, the correlation between our unsupervised quality test and gold standard based tagging quality measures is high. For the English WSJ corpus, the Q(ZCC) algorithm gives better results than CT with probability 82-100% (depending on the external quality measure used). Q(CT) is shown to be better than the original CT algorithm as well. Our results are better in most evaluation measures than all previous results reported in the literature for this task, and are always better than Clark's average results.

References

[1] Alexander Clark, "Combining Distributional and Morphological Information for Part of Speech Induction.", *EACL '03*.

[2] Noah A. Smith and Jason Eisner, "Contrastive Estimation: Training Log-Linear Models on Unlabeled Data.", *ACL '05*.

[3] Sharon Goldwater and Tom Griffiths,, " A fully Bayesian approach to unsupervised part-of-speech tagging.", *ACL '07*.

[4] Mark Johnson, "Why Doesnt EM Find Good HMM POS-Taggers?", *EMNLP-CoNLL '07*.

Using Synonyms for Arabic-to-English Example-Based Translation

Kfir Bar and Nachum Dershowitz
School of Computer Science, Tel Aviv University, Ramat Aviv, Israel
{kfirbar,nachumd}@post.tau.ac.il

We have developed an experimental Arabic-to-English example-based machine translation (EBMT) system, which exploits a bilingual corpus to find examples that match fragments of the input source-language text Modern Standard Arabic (MSA), in our case—and imitates its translations. Translation examples were extracted from a collection of parallel, sentence-aligned, unvocalized Arabic-English documents, taken from several corpora published by the Linguistic Data Consortium. The system is non-structural: translation examples are stored as textual strings, with some additional inferred linguistic features.

In working with a highly inflected language, finding an exact match for an input phrase with reasonable precision presumably requires a very large parallel corpus. Since we are interesting in studying the use of relatively small corpora for translation, matching phrases to the corpus is done on a spectrum of linguistic levels, so that not only exact phrases are discovered but also related ones. In this work, we looked in particular at the effect of matching synonymous words.

To explore the possibility of matching fragments based on source-language synonyms, we created a thesaurus for Arabic, organized into levels of perceived synonymy. Since an Arabic WordNet is still under development, we developed an automatic technique for creating a rough thesaurus, based on English glosses provided with the Arabic stem list of the Buckwalter morphological analyzer. To create a thesaurus of nouns, we looked at the English WordNet synsets of every English translation of a stem in the Buckwalter list. A synset containing two or more of the translations is taken to be a possible sense for the given stem. This assumption is based on the idea that if a stem has two or more different translations that semantically intersect, it should likely be interpreted as their common meaning. We also considered WordNets hyponym-hypernym relations between the translations senses, and take a stem to have the sense of the shared hyponym. Different strengths of synonymy were defined according to the closeness and uniqueness of these relations. The quality of the systems resultant translations were measured for each of the different levels of synonymy.

In the matching step, the system uses various levels of morphological information to broaden the quantity of matched translation examples and to generate new translations based on morphologically similar fragments. All the Arabic translation examples were morphologically analyzed using the Buckwalter morphological analyzer, and then part-of-speech tagged using AMIRA, in such a way that, for each word, we consider only the relevant morphological analyses with the corresponding part-of-speech tag. For each Arabic word in the translation example, we look up its English equivalents in a lexicon created from the Buckwalter glossaries, and also expand those English words with synonyms. Then we search the English version of the translation example for all instances of these words at the lemma level, creating an alignment table containing one-to-one alignment entries. In addition, several special alignment cases are handled. For instance, an English noun-phrase that contains unaligned words is usually combined with its aligned words, if any, creating a one-to-many entry in the alignment table. In this way, most of the prepositions, definite articles and indefinite articles are covered. Another special case is connecting the immediate noun of an aligned verb to its equivalent.

Demarcating noun-phrase boundaries and obtaining part-of-speech information for the English part is accomplished using Brills part-of-speech tagger and the BaseNP chunker, respectively.

The Arabic version of the corpus was indexed on the word, stem and lemma levels (stem and lemma, as defined by the Buckwalter analyzer). So, for each given Arabic word, we are able to retrieve all translation examples that contain that word on any of those three levels.

In using synonyms for matching, we also considered the relevance of the subject matter of translation examples to any given input sentence. Topics were determined using a classifier that was first trained on the English Reuters training corpus and then used for classifying the English part of the translation examples in our parallel corpus. With this classification of the samples in hand, we trained an Arabic-language classifier on the Arabic version of the parallel corpus, which was then used to classify new Arabic input documents.

During the transfer step, matched fragments are translated using the English version of the parallel corpus. Currently, the system translates each fragment separately and then concatenates those translations to form an output target-language sentence, preferring longer translated fragments, since the individual words appear in a larger context. Recombining those translations into a final, coherent form is left for future work.

We found that synonyms benefit from being matched carefully by considering the context in which they appear. Comparing other ways of using context to properly match the true senses of ambiguous synonyms is definitely a direction for future investigation.

Another interesting observation is the fact that using synonyms on a large corpus did not result in any improvement of the final results, as it did for the smaller corpus. This suggests that synonyms can contribute to EBMT for resource-poor languages other than Arabic, by enabling the system to better exploit the small number of examples in the given corpus.

A Hebrew-to-Arabic Syntax-based Machine Translation System

Reshef Shilon
Department of Linguistics, Tel Aviv University and
Department of Computer Science, University of Haifa
reshefs1@post.tau.ac.il

Introduction The dominant paradigm in contemporary machine translation [Brown et al.1990] relies on large-scale parallel corpora from which correspondences between the two languages can be extracted. However, such abundant parallel corpora only exist for few language pairs. Hebrew and Modern Standard Arabic (MSA) share many lexical, morphological, syntactic and semantic similarities, being both Semitic, but they are still not mutually comprehensible. We describe work in progress whose goal is to construct a Hebrew-to-Arabic machine translation system, using the STAT-Xfer framework [Lavie2008], which is suited for low-resource language pairs such as Hebrew-Arabic. In this framework, a transfer lexicon and a manually crafted grammar, which contains rules that map constituent structures using rich syntax, are used to create a lattice of hypotheses. A statistical decoder searches the hypotheses space for an optimal solution, according to a language model and machine-learned parameters. This framework is currently being applied for building a Hebrew-to-English MT system. Our work relies on many results and outcomes learned from the H2E MT work.

Resources We use the following resources:

a. A bilingual dictionary with a reasonable coverage, with no statistical weights.
b. A morphological analyzer for Hebrew [Itai and Wintner2008].
c. A morphological generator for Arabic [Habash2004].
d. A tokenized version of the Arabic GigaWord corpus as a language model

Challenges Arabic, being a morphologically-rich language (like Hebrew), presents many challenges in word and sentence generation. These challenges are expressed across the lexical, morphological, syntactic and computational levels. Some examples follow.

In many cases, an Arabic verb requires a different set of prepositions from its Hebrew counterpart. This presents a challenge in properly creating Arabic long-distance agreement (with respect to the correct preposition) between the verb and its argument.

Many syntactic challenges stem from correctly forcing agreement. There is rich agreement on many features between different constituents, such as between V-Subj, V-O and N-Adj. Arabic also displays some surprising agreement constraints, such as the plural form of irrational (non-human) nouns. Any such noun is treated as 3FS, regardless of the original gender. This is relatively easy to deal with in local contexts, but harder in long-distance agreement.

(1)

Al+AqlAm	*Alty*	*A$trA+hA*	*Al+wld*	*Ams*	*jmylp*
pen-m.pl.def	that.f.sg	buy-past.3.m.sg+she-acc.	boy-m.sg.def	yesterday	pretty-f.sg.indef

'The pens which the boy bought yesterday are pretty'

Another surprising fact is V-Subj number agreement, where the verb is always in singular form when it preceeds the subject, and agrees with the subject on number when the verb succeeds the subject.

Another syntactic challenge is different word order. In Arabic the predominant word order is VSO, but there are many SVO sentences. The major implication of this is greater difficulty in forcing correct agreement between the verb and the possibly distant object, since the LM is less effective in such cases.

A computational challenge we are facing is the exponential explosion of the lattice, which is exemplified by the number of possible morphological froms each Arabic word can have (109 surface forms for verbs, 72 for nouns).

Solutions We use the rich syntax provided by the transfer rules to map corresponding structures and local dependencies across the two languages. We account for more distant dependencies by applying local lexical agreement features to larger constituents. Agreement issues are all treated in the manually crafted grammar, while in some cases we generate several possible hypotheses and let the LM choose the correct option.

Despite the challenges listed above, there are many similarities between the languages that make the translation process easier. Such similarities include similar lexical features, similar morphology and word structures,and similar syntactic structures All of these similarities make our task easier to handle.

Preliminary Results While we still do not have robust evaluation results, we provide an example translation of a simple phrase to demonstrate the capabilities of the system. We compare our results with Googles Hebrew-to-Arabic MT system [1].

(2) (a) *hncigim* *šlkm* *nkxw* *bišibh*
representative.pl.m.def you.pl.m.poss attend.past.3.pl in+meeting.sg.f.def
'your representatives attended the meeting'

(b) *HDr* *mmvlwkm* *Aljlsp*
attend.past.sg.m representative.pl.m.nom+you.pl.m.poss meeting.def
'your representatives attended the meeting' (Stat-XFER)

(c) *wmmvlwkm* *AlHADryn* *fy* *AlAjtmAE*
and+representative.pl.m.nom+you.pl.m.poss attend.participle.pl.m.def.acc/gen in meeting.def
'And your representatives that attended the meeting' (Google)

Example (2 b) demonstrates correct translation of the preposition, differing word order, V-Subj number agreement in Arabic, and conversion of a possessive construction using *šl* from Hebrew to *Idafa* in Arabic. In example (2 c), Google fails on translating the Hebrew verb correctly, enforcing case, and the correct choice of preposition (*HDr* requires a direct object).

References

[Brown et al.1990] P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

[Habash2004] Nizar Habash. 2004. Large scale lexeme based arabic morphological generation. In *Proceedings of Traitement Automatique du Langage Naturel (TALN-04)*, Fez, Morocco.

[Itai and Wintner2008] Alon Itai and Shuly Wintner. 2008. Language resources for Hebrew. *Language Resources and Evaluation*, 42:75–98, March.

[Lavie2008] Alon Lavie. 2008. Stat-XFER: A general search-based syntax-driven framework for machine translation. In Alexander F. Gelbukh, editor, *CICLing*, volume 4919 of *Lecture Notes in Computer Science*, pages 362–375. Springer.

[1] `http://www.google.com/language_tools`, accessed May 5th, 2010.

SSA – A Hybrid Approach to Sentiment Analysis of Stocks

Ronen Feldman; Benjamin Rozenfeld; Micha Y. Breakstone

Digital Trowel
Airport City, ISRAEL

In recent years, as the content on the web is becoming more and more user-generated, the challenge posed by Sentiment Analysis (SA) has become ever more relevant, and its applications ever more widespread and potentially valuable. Gleaning not only the objective data, but also the sentiment or opinion associated with a text passage, is nowadays crucial for any NLP system that hopes to extract a coherent picture of the semantic space under investigation. Applications of SA range from ascertaining customer satisfaction with products such as pharmaceutical drugs based on health forums, to analyzing the sentiment associated with stocks and companies as conveyed in news sites, financial editorials and blogs (the latter being the chief concern of this paper).

Although SA is considered extremely important and relevant, the technological challenge it poses has proved quite formidable. The problem of implementing an accurate SA system has two aspects, a technical aspect and a linguistic one. The technical challenge lies in the fact that a valuable SA system must be able to sift through extremely large corpora and produce accurate results within seconds. In this paper we do not delve into this aspect, but merely report in subsequent sections on the results and capabilities of the Stock Sentiment Analysis (SSA) system. The linguistic aspect of the problem is of main concern to us herein.

To appreciate how evasive the linguistic problem of ascertaining sentiment can be, consider the following pet-example.

A-priori the word *"great"* would probably be judged as conveying positive sentiment, but when followed by the word *"disaster"* to form the phrase *"great disaster"*, this judgment is obviously reversed. In addition, when appearing in a complex sentence such as: *"Apple's release of the iPad may prove a* great disaster *for Barnes & Nobles who just released their new Nook reader"*, a further complication of "sentiment ownership", so to speak, arises. However, even if these challenges are overcome, it quickly becomes clear that the problem transcends the lexical and phrasal analysis and enters the domain of semantics and pragmatics. Consider the following sentence: *"Such*great fiascos *as Toyota has suffered over the past months often ultimately prove to be a blessing in disguise."* Or, on a different note: *"Microsoft's previous success to 'see the view' could not forestall the* great fiasco *of Vista."*

From the above set of examples, we learn that the challenge in question cannot be solved on a lexical and phrasal level alone, even if compositional factors such as polarity-reversers, modals, counterfactuals and speculative indicators are taken into account. Our hybrid technological approach as implemented in the SSA system begins with the observation that sentiment is conveyed on three intertwined levels of increasing structural complexity, namely the lexical, phrasal and semantic-pragmatic levels of structure.

The paper is structured as follows. We begin with a survey of recent papers covering the current approaches and implementations of SA systems. We then describe the architecture of our SSA system, devoting our attention to the 3 core linguistic components: Dictionary (lexical) Based SA, Pattern Based SA and Event Based SA. In doing so, we show how interweaving these 3 components enables us to account for sentiment as conveyed on all linguistic levels, and in particular allows the SSA to be sensitive to syntactic, semantic and pragmatic effects. Finally we present an experimental evaluation, which not only validates the superiority of the SSA's accuracy but also serves to prove that a hybrid approach is indeed crucial for obtaining such results.

Semi-Supervised Recognition of Sarcastic Sentences in Twitter and Amazon

Dmitry Davidov, Oren Tsur and Ari Rappoport
ICNC, School of Computer Science and Engineering
The Hebrew University

Sarcasm (also known as *verbal irony*) is a sophisticated form of speech act in which the speakers convey their message in an implicit way. One inherent characteristic of the sarcastic speech act is that it is sometimes hard to recognize. The difficulty in recognition of sarcasm causes misunderstanding in everyday communication and poses problems to many NLP systems such as online review summarization systems and dialogue systems due the failure of state of the art sentiment analysis systems to detect sarcastic comments. In this paper we present a robust algorithm for automatic identification of sarcastic sentences.

One definition for sarcasm is: *the activity of saying or writing the opposite of what you mean, or of speaking in a way intended to make someone else feel stupid or show them that you are angry* (Macmillan English Dictionary). Using the former definition, sarcastic utterances appear in many forms. It is best to present a number of examples which show different facets of the phenomenon, followed by a brief review of different aspects of the sarcastic use. The sentences are all taken from our experimental data sets (data set is indicated in parenthesis):

1. *"thank you Janet Jackson for yet another year of Super Bowl classic rock!"* (Twitter)
2. *"He's with his other woman: XBox 360. It's 4:30 fool. Sure I can sleep through the gunfire"* (Twitter)
3. *"Wow GPRS data speeds in Bedford are blazing fast."* (Twitter)
4. *"twitter is down, nobody expected that."* (Twitter)
5. *"[I] Love The Cover"* (book, amazon)
6. *"Great for insomniacs"* (book, amazon)
7. *"Defective by design"* (music player, amazon)

Example (1) refers to the supposedly lame music performance in super bowl 2010 and attributes it to the aftermath of the scandalous performance of Janet Jackson a few years earlier. Note that the previous year is not mentioned and the reader has to guess the context (use universal knowledge). The sarcastic marker is the word *yet*. (2) is composed of three short sentences, each of them sarcastic on its own. However, combing them in one tweet brings the sarcasm to its extreme. Example (3) is a factual statement without explicit opinion. However, having fast connection is a positive thing. The sarcasm emerges from the clear falseness (semantic value) combined with over exaggeration ('wow', '*blazing*-fast'). Twitter servers suffer frequent downtimes, as reflected in (4). As in (3) the sentence does not convey positive or negative sentiment explicitly.

Example (5) from Amazon, might be a genuine compliment if it appears in the body of the review. However, recalling the expression 'don't judge a book by its cover', choosing it as the title of the review reveals its sarcastic nature. (6) conveys a clear positive sentiment ('great'). The sarcasm requires world knowledge (insomnia vs. boredom $\mapsto$ sleep). Although the negative sentiment is very explicit in the iPod review (7), the sarcastic effect emerges from the pun that assumes the knowledge that the design

is one of the most celebrated features of Apple's products. (None of the above reasoning was directly introduced to our algorithm.)

As sarcasm is a sophisticated form of speech act often misunderstood by the hearer, modeling the underlying patterns of sarcastic utterances is interesting from the psychological and cognitive perspectives. Recognition of sarcasm can also benefit various NLP systems such as review summarization and dialogue systems. Following the 'brilliant-but-cruel' hypothesis, it can help improve ranking and recommendation systems. All systems currently fail to correctly classify the sentiment of sarcastic sentences.

In this work we present SASI, a Semi-supervised Algorithm for Sarcasm Identification. The algorithm employs two modules: (I) semi supervised pattern acquisition for identifying sarcastic patterns that serve as features for a classifier, and (II) a classification algorithm that classifies each sentence to a sarcastic class.

We tested our algorithm on two data sets: a collection of 6 million tweets from Twitter and a collection of 70000 user reviews from Amazon. The two data sets radically differ from each other. Our algorithm performed well in both domains, substantially outperforming a strong baseline based on semantic gap and a second algorithm that employs the *#sarcasm* hashtag for supervised learning. To further test its robustness we also trained the algorithm in a cross domain manner, achieving good results.

Driving Language Interpretation from the World's Response

Ming-Wei Chang, James Clarke, Dan Goldwasser and Dan Roth
University of Illinois at Urbana Champaign
{mchang21,clarkeje,goldwas1,danr}@illinois.edu

Semantic parsing, the process of converting text into a formal meaning representation (MR), is one of the key challenges in natural language processing. Unlike shallow approaches for semantic interpretation (e.g., semantic role labeling and information extraction) which often result in an incomplete or ambiguous interpretation of the natural language (NL) input, the output of a semantic parser is a complete meaning representation that can be executed directly by a computer program. Current semantic parsing works concentrate on providing natural language interfaces to computer systems, for example — natural language access to databases. In these settings the question posed in natural language is converted into a formal database query that can be executed to retrieve information. The following pair is an example of a NL input query and its corresponding meaning representation.

Example 1 *Geoquery input text and output MR*
"What is the largest state that borders Texas?" $\Rrightarrow$ `largest(state(next_to(const(texas))))`

Existing works employ supervised machine learning techniques to construct a semantic parser. The learning algorithm is given a set of input sentences and their corresponding meaning representation, and learns a statistical semantic parser — a set of rules mapping lexical items and syntactic patterns to their corresponding meaning representation and a score associated with each rule. Given a sentence, these rules are applied recursively to derive the most probable meaning representation. Since semantic interpretation is limited to syntactic patterns identified in the training data, the learning algorithm requires considerable amounts of annotated data to account for the syntactic variations associated with the meaning representation. Annotating sentences with their corresponding MR is a difficult, time consuming task; minimizing the supervision effort required for learning is a major challenge in scaling semantic parsers.

We propose a new model and learning paradigm for semantic parsing aimed to alleviate the the supervision bottleneck. Following the observation that the target meaning representation is to be executed by a computer program which in turn provides a response or outcome; we propose a *response driven learning framework* capable of exploiting feedback based on the response. The feedback can be viewed as a teacher judging whether the execution of the meaning representation produced the desired response for the input sentence. This type of supervision is very natural in many situations and requires no expertise thus can be supplied by any user.

Continuing with Example 1, the response generated by executing a database query would be used to provide feedback. The feedback would be whether the generated response is the correct answer for the input question or not, in this case *New Mexico* is the desired response.

In *response driven semantic parsing*, the learner is provided with a set of natural language sentences and a feedback function that encapsulates the teacher. The feedback function informs the learner whether its interpretation of the input sentence produces the desired response. We consider scenarios where the feedback is provided as a binary signal, correct $+1$ or incorrect -1.

The new form of supervision poses a challenge to conventional learning methods: semantic parsing is in essence a structured prediction problem requiring supervision for a set of interdependent decisions, while the provided supervision is binary, indicating the correctness of a generated meaning representation. To bridge this difference we propose two novel learning algorithms adapted for the response driven setting.

Furthermore, to account for the many syntactic variations associated with the MR, we propose a new model for semantic parsing that allows us to learn effectively and generalize better. We model semantic interpretation as a sequence of interdependent decisions, mapping text spans to predicates and use syntactic information to determine how the meaning of these logical fragments should be composed. We frame this process as an Integer Linear Programming (ILP) problem, a powerful and flexible inference framework that allows us to inject relevant domain knowledge into the inference process, such as specific domain semantics that restrict the space of possible interpretations.

We evaluate our learning approach and model on the well studied Geoquery domain, a database consisting of U.S. geographical information, and natural language questions. Our experimental results show that using our model with response driven learning we can outperform existing models trained with annotated logical forms.

Exploiting Synonym Choice to Identify Discrete Components of a Document

Navot Akiva, Dept. of Computer Science, Bar-Ilan University, Ramat Gan
Idan Dershowitz, Dept. of Bible, Hebrew University, Jerusalem
Moshe Koppel, Dept. of Computer Science, Bar-Ilan University, Ramat Gan
{navot.akiva,dershowitz,moishk}@gmail.com

When studying ancient texts, scholars often contend with documents that appear to be composite. A key challenge is to tease apart the various constituents.

One notable example of such a text is the Pentateuch, in which many scholars have found what they think are discrete narrative threads. The most prominent theory relating to the literary history of the Pentateuch is known as the "Documentary Hypothesis."

In some cases, scholars have at their disposal several widely divergent manuscripts, providing them with valuable data to better approach the primary text or texts. But when the available manuscripts are less obliging, the work of analyzing composite texts is generally done in an impressionistic fashion. Factors such as repetitions, contradictions, or possible interruptions in narrative flow, play a large part in scholars' considerations. But what is for one scholar an intolerable repetition or contradiction, is for another an instance of sophisticated literary variation. We propose to set this work on a firm algorithmic basis by identifying an optimal stylistic sub-division of a given manuscript. We do not concern ourselves with how or why such distinct threads might exist.

The most straightforward way to divide a potentially composite document is to represent segments of text as numerical vectors reflecting the frequencies of lexical features and to use clustering algorithms to find natural clusters. However, this method tends to divide texts topically rather than stylistically. Limiting features to function words is inadequate, and in tests on the books of Jeremiah and Ezekiel, we find that clustering on function words fails to separate out the two books.

Our main innovation is the use of synonym choice. Our hypothesis is that different literary works should differ in the proportions with which different synonyms in the same synset are found. By focusing our attention only on words that have synonymous counterparts in the same set of books, we can be relatively confident that the resulting division will not be according to topic. If one author speaks of a "big" house and another of a "large" one, the difference between the two is not subject matter, but personal preference or style.

We leverage very precise translations of the Bible as well as manual sense tagging for the Bible to automatically identify sets of synonyms. The automatically generated synonym set list is then manually cleaned of obvious errors. We also use a specially designed similarity measure that captures the extent to which different passages make similar/different synonym choices. This method separates Jeremiah and Ezekiel very well.

There is one additional hurdle that must be handled. Initially, we used the standard chapters as our natural units. But these units may not be pure; a single chapter might be a mix of two or more literary strands. Thus, we develop several new algorithms for automatically identifying literary boundaries.

Results show that optimal separation of the Pentateuch into two clusters roughly correlates with the portions identified by Bible scholars as P and non-P.

Irony interpretation: Will expecting it make a difference?

Rachel Giora
Linguistic
Tel Aviv University
giorar@post.tau.ac.il
http://www.tau.ac.il/~giorar

Results from 10 experiments support the view that, regardless of strength of context, when an end-product interpretation of an utterance does not rely on the salient (lexicalized and prominent) meanings of its components, it will not be faster than nor as fast to derive as when it does. To test this view, we examined the interpretations of salience-based (here, literal) interpretations and context-based (here, ironic) interpretations, at different temporal stages, in strong contexts inducing an expectation for irony. In Experiment 1, we first tested the claim that an "ironic situation" is a strong context, inducing an expectation for an ironic utterance (Gibbs, 2002). Results, however, show that in both an "ironic situation" as well as in a nonironic situation, readers preferred a literal ending over an ironic ending. In Experiment 2 we tested the hypothesis that an "ironic situation" facilitates irony interpretation (Gibbs, 2002). Results show that this is not the case: an "ironic situation" did not facilitate an ironic utterance compared to a nonironic situation (Giora et al., 2009). In Experiment 3, expectancy was manipulated by introducing an ironic speaker in vivo who also uttered the target utterance. Findings show that ironic targets were slower to read than literal counterparts. Experiment 4 shows that ironies took longer to read than literals and that response times to ironically related probes were longer than to literally related probes, regardless of context bias. Experiments 5 and 6 show that, even when participants were allowed long processing times (750 ms and 1000 ms ISIs respectively) and were exclusively presented ironically biasing contexts, the expectancy for irony acquired throughout such exposure did not facilitate expectancy-based compared to salience-based interpretations (Giora et al., 2009). Replication of Experiments 5-6 in Experiments 7-8, in which an attempt was made at further strengthening contextual expectation for irony, did not change the pattern of results; even when participants were told we were investigating irony interpretation, ironies were still slower to interpret than salience-based literal interpretations. In Experiment 9 participants were presented the same items as in Experiments 5-8. As before, they were informed about the aim of the experiment but, in addition, were allowed even longer processing times (of 1500 ms ISI). Still, pattern of results did not change; salience-based (literal) utterances were always processed faster, with context-based (ironic) interpretations lagging behind. Experiment 10 replicated experiment 9 with even a longer ISI (of 2000 ms). Even at this temporal stage, salience-based interpretations were always processed first.

References

Giora, Rachel, Ofer Fein, Ronie Kaufman, Dana Eisenberg, and Shani Erez. (2009). Does an "ironic situation" favor an ironic interpretation? In G. Brône & G. Vandaele (Eds.) *Cognitive Poetics.* Applications of Cognitive Linguistics series, 383-399. Mouton de Gruyter.

Giora, R., Fein, O., Laadan, D., Wolfson, J., Zeituny, M., Kidron, R., Kaufman, R., & Shaham, R. (2007). Expecting irony: Context vs. salience-based effects. *Metaphor and Symbol, 22,* 119-146.