АВТОМАТИЧЕСКОЕ СНЯТИЕ МОРФОЛОГИЧЕСКОЙ ОМОНИМИИ В КОРПУСАХ НОВОГРЕЧЕСКОГО ЯЗЫКА И ЯЗЫКА ИДИШ

Кузьменко Е. А. (eakuzmenko_2@edu.hse.ru),
Мустакимова Э. Г. (egmustakimova_2@edu.hse.ru)

Национальный исследовательский университет «Высшая школа экономики», Москва, Россия

Ключевые слова: морфологический анализ, снятие омонимии, корпусная лингвистика, греческий язык, язык идиш

AUTOMATIC DISAMBIGUATION IN THE CORPORA OF MODERN GREEK AND YIDDISH

Kuzmenko E. A. (eakuzmenko_2@edu.hse.ru),
Mustakimova E. G. (egmustakimova_2@edu.hse.ru)

National Research University Higher School of Economics, Moscow, Russia

The problem of morphological ambiguity is widely addressed in the modern NLP. Mostly ambiguity is resolved with the use of large manually-annotated corpora and machine learning. However, such methods are not always available, as good training data is not accessible for all languages. In this paper we present a method of disambiguation without gold standard corpora using several statistical models, namely, Brill algorithm (Brill 1995) and unambiguous n-grams from the automatically annotated corpus. All the methods were tested on the Corpus of Modern Greek and on the Corpus of Modern Yiddish.

As a result, more than a half of words with ambiguous analyses were disambiguated in both corpora, demonstrating high precision (>80%). Our method of morphological disambiguation demonstrates that it is possible to eliminate some of the ambiguous analyses in the corpus without specific linguistic resources, only with the use of raw data, where all possible morphological analyses for every word are indicated.

Keywords: morphological tagging, morphological disambiguation, corpus linguistics, Modern Greek, Yiddish

1. Introduction

As the usage of corpus methods becomes widespread in linguistics, the problem of ambiguity in existing corpora turns out to be more and more significant. To perform deep linguistic analysis a researcher needs language data of high quality. Meanwhile, morphological processing of the corpus data involves two steps: assigning morphological analyses to tokens and, as wordforms in a language are often ambiguous, disambiguation. Ambiguity in corpora does not allow linguists to make detailed queries and get exact results because they receive a lot of irrelevant data. Disambiguation by hand is time-consuming; therefore, it is essential that ways of computer-aided disambiguation are developed. Most of the disambiguation techniques are based on the implementation of machine learning. Machine learning implies having a huge manually disambiguated corpus, which can be used for training the algorithm. However, such resources are not available for every corpus and every language. For this reason we need to find an approach to disambiguate texts with no knowledge about the statistics of word occurrences in particular contexts and with no manual annotation. Despite these rough demands, they should show high accuracy and amplitude.

In this paper we consider automatic disambiguation techniques for the corpora of Modern Greek and Yiddish, which do not have pre-disambiguated subcorpora. We combine several existing disambiguation algorithms in a more effective way, adapt POS-tagging algorithms to the disambiguation problems (Brill 1995) and develop a technique of our own (disambiguation on the basis of unambiguous n-grams found in the corpus). We estimate the effectiveness of each technique and compare them.

The originality of our work lies in absence of any manually processed data. Moreover, we work with morphologically rich languages. All the data available to us is raw corpora in which every word is assigned all possible morphological analyses.

2. Corpora

2.1. The corpus of Modern Greek

The corpus of Modern Greek[1] consists of 26 million tokens. The majority of texts come from Greek newspapers and belong to the 21st century. Also there are such genres as fiction (both native and translated works), poetry, publicistic writing and scientific literature. These texts belong to 19th–21st centuries. The Corpus of Modern Greek is based on the EANC platform (Arkhangelskiy et al. 2013). Morphological information in this corpus is stored according to the UniParser standard.

Every word is assigned all possible analyses; for example, the word occurence μέσα could be assigned the following analyses:

1. μέσα, ADV, “inside”;
2. μέσο, NOUN,n,pl,acc, “medium”;
3. μέσο, NOUN,n,pl,nom, “medium”;

[1] http://web-corpora.net/GreekCorpus/search/

4. μέσος, ADJ,pos,n,pl,acc, “middle”;
5. μέσος, ADJ,pos,n,pl,nom, “middle”.

Before performing disambiguation, we estimated baseline parameters of ambiguity in our corpus (Table 1):

Table 1. Baseline parameters for ambiguity in the corpus

Number of tokens	Percentage of ambiguous words	Ambiguity rate
26,075,298	43%	1.64

In this table the parameters signify the following:

- Number of tokens—number of words in the corpus
- Percentage of ambiguous words—the ratio of tokens which have more than one analysis to the overall number of tokens in the corpus
- Ambiguity rate—the ratio of all tags in the corpus to all tokens

Most of the words had 2 or 3 analyses, and sometimes they had 4 or even 5 analyses.

Overall, there were almost 10 thousand (9,987, to be exact) different types of ambiguity, and there were 11 thousand (10,842) different ambiguous word instances.

The most frequent types of ambiguity were the following (different morphological analyses are separated with dashes):

1. το,ART,n,sg,acc—το,ART,n,sg,nom—το,PRO,n,sg,acc—τον,ART,m,sg,acc
2. του,ART,m,sg,gen—του,ART,n,sg,gen—του,PRO,m,sg,gen—του,PRO,n,sg,gen
3. του,ART,m,sg,gen—του,ART,n,sg,gen
4. είμαι,V,pres,3,pl—είμαι,V,pres,3,sg
5. με,PR—με,PRO,1p,sg,acc
6. της,ART,f,sg,gen—της,PRO,f,sg,gen
7. των,ART,pl,gen—των,PRO,pl,gen
8. την,ART,f,sg,acc—την,PRO,f,sg,acc
9. είμαι,V,past,3,pl—είμαι,V,past,3,sg
10. τα,ART,n,pl,acc—τα,ART,n,pl,nom—τα,PRO,n,pl,acc

These 10 types of ambiguity out of 10 thousand overall together constitute 15% of ambiguity in the corpus.

2.2. The corpus of Modern Yiddish

The Corpus of Modern Yiddish[2] (CMY) is a joint project of the Russian Academy of Sciences and the University of Regensburg, which started in 2007. The corpus comprises mainly publicistic texts, fiction is represented to a much lesser degree. For now the volume of the CMY is about 4 million tokens.

[2] http://web-corpora.net/YNC/search/

As in the Corpus of Modern Greek, each word in the CMY is supplied with a list of all possible morphological interpretations. The ambiguity rate in Yiddish is even higher than in Greek. Baseline parameters for CMY are shown in Table 2.

Table 2. Baseline parameters for ambiguity in the Corpus of Modern Yiddish

Number of tokens	Percentage of ambiguous words	Ambiguity rate
4144524	39,5%	2.026

In the case of CMY it is impossible to resolve all cases of ambiguity. The first reason is that almost all nouns are supplied with at least 4 analyses. A noun in Yiddish has 4 cases (nominative, genitive, dative, accusative), but the case forms look identical for most nouns. Since complete resolution for all nouns is impossible and partial resolution would result in inconsistent markup and inconvenient corpus search, such cases of ambiguity will be ignored. The second reason is that verbs can merge with pronouns into one word and dative case prepositions merge with definite articles. Such merges are supplied with at least two tags. Thus, despite the fact that we want to map each token in the corpus to a single morphological interpretation, we have to accept multiple analyses for nouns and merged wordforms.

The corpus has 729 types of different combinations of tags in ambiguous words and about 24,000 different words that are homonymous. Observe that the corpus has about 2 million ambiguous words in total, and only 24 thousand different ambiguous words. According to Zipf's Law and these numbers, it is logical to assume that resolving some small amount of the most frequent homonymy types should significantly lessen the amount of ambiguity.

3. Related work

We are not the first to apply data-driven algorithms to the task of morphological disambiguation. It has been already done for such languages as Icelandic, Swedish and Turkish. The researchers working on these languages employed the Brill algorithm, and so did we. Similarly to our decision, this algorithm is not applied solely, but in combination with other approaches, such as composing linguistic rules and using n-grams. The results for other languages are the following: for Icelandic the precision of 93.65% was achieved (Helgadóttir 2004). For Swedish the results are slightly worse—only 84.5% precision (Maurier et al. 2003). For Turkish, on the contrary, significant results are reported—the authors managed to achieve the precision of 96.8% (Sak et al. 2007). Maybe this result is due to the morphology of Turkic languages, which is more easily formalized compared to languages with cumulative morphology.

Some research has been done specifically for the Greek language. However, corpora of the Greek language are not numerous: there are such corpora as HNC (Hatzigeorgiu et al. 2000), DELOS (Kermanidis et al 2002), and CGT (Goutsos 2010). Meanwhile, these corpora do not provide morphological disambiguation, or it is of poor quality. There were also some attempts to design tools for disambiguation in the Greek

language, for example, the research described in (Petasis 1999). However, in this case the tagset is very limited, so no detailed morphological information is provided. Also this approach uses a pre-disambiguated part of the corpus that serves as a golden standard. Therefore, our work is very different from the previous research because we, as it was already stated, do not use manually processed data.

If we talk about Yiddish, there are three written corpora of Yiddish: the Aston corpus of Soviet Yiddish which does not have morphological annotation, Yiddish Treebank of the University of Pennsylvania (no one knows how exactly it was annotated, probably manually) and the CMY, which is a comprehensive, annotated and a freely available corpus. Also the Yiddish language lacks tools for disambiguation, so we can not compare the result of the task with the works of previous researchers.

As we can see, Greek and Yiddish can be called under-resourced languages to the full extent: there are not many corpora for these languages and disambiguation in these corpora was not properly performed. This means that Greek and Yiddish need a method for disambiguation which would not require linguistic resources and will provide high quality despite these constraints.

4. Methods

Our decision was to find the way to combine data-driven and rule-based algorithms[3]. We used the following data-driven methods:

- trasformation-based error-driven learning (Brill 1995a; Brill 1995b);
- using data about bigrams and trigrams in which the word under consideration can be found;
- the user interface for disambiguating based on bigrams and trigrams.
  Also we used the hand-crafted rules approach.

For evaluation of the methods we used a testing part of the Greek Corpus which contains 866,091 tokens. In the case of Yiddish we used the whole CMY as a test corpus since its volume is fairly small.

4.1. The Brill algorithm

Transformation-based error-driven algorithm for pos-tagging and disambiguation purposes was developed by Eric Brill in 1995. This algorithm is very useful in our situation because it is unsupervised (which means that we do not need the disambiguated corpus). We can achieve significant results by just using unambiguous word instances from our corpus.

We tested two versions of Brill disambiguation algorithm:

---

[3] In (Halperen et al. 2001) it was shown that taggers combining several approaches result in a higher accuracy.

1. The version that executed only disambiguation with respect to the part of speech (the cases where words had analyses with different POS-tags were resolved)
2. The version that executed full disambiguation (the cases where words had analyses with the same POS-tag, but different values, were also resolved)

After applying the first version of the Brill algorithm to the test corpus the ambiguity parameters changed in the following way (Table 3):

Table 3. Ambiguity parameters of the test corpus after POS-disambiguatuion by the Brill algorithm

Corpus	Number of tokens	Percentage of ambiguous words	Ambiguity rate
Greek	866,091	29.0%	1.36
Yiddish	4,144,524	35.1%	1.86

Also we applied the second version of the Brill algorithm to the testing corpus of Greek, and the ambiguity parameters changed in the following way (Table 4):

Table 4. Ambiguity parameters of the testing corpus after full disambiguation by the Brill algorithm

Corpus	Number of tokens	Percentage of ambiguous words	Ambiguity rate
Greek	866,091	37%	1.57

In the case of Yiddish, the quality of POS-disambiguation performed by the Brill algorithm was fair enough. However, the picture for the Greek language was different: the first version of the Brill algorithm works extensively (recall ~41.4%), but most words are changed incorrectly (precision ~8.2%). The second version of the algorithm, however, shows high precision (~74%), but changes very few words (recall ~8.7%). This results in similar values of F_1-score (13.73 in the first case and 15.62 in the second case). The similarity of the scores for the methods shows that they are almost equally effective (or, in our case, ineffective).

Then we tested the results of the algorithm when we first applied the Brill algorithm in the full mode, and then finished disambiguation by the POS-version. The idea was that after applying the first variant of the algorithm the number of unambiguous words increases and drives the second version to be more accurate. The results are displayed in Table 5:

Table 5. Ambiguity parameters and effectiveness measures after applying two versions of the Brill algorithm

Corpus	Number of tokens	Percentage of ambiguous words	Ambiguity rate	Precision (%)	Recall (%)	F_1-score
Greek	866,091	26%	1.32	22.7	49	31.02

As we can see from the table above, POS-disambiguation by the Brill algorithm indeed becomes more effective when applied to the pre-processed data. Its precision increases, though it is still on the low level, and the F_1-score shows that such results are more valuable than the previous results (31.02 compared to previous ~14). This experiment shows that POS-disambiguation by the Brill algorithm can serve as the final step in the process of disambiguation, when it would become more effective.

4.2. Bigrams and trigrams

This approach is similar to probabilistic Markov models described in (Kupiec 1992), but it works in a different and simpler way. Training a good bigram model requires a manually annotated corpus which we do not have. For this reason, we decide to automatically extract non-ambiguous parts of the corpus and treat it as an etalon. Non-ambiguous bigrams are those both words of which have no ambiguity.

In the Python programming language, the bigram model is realized as a dictionary where each key is a morphological tag and the corresponding value is an array of tuples. Each tuple contains a) a tag that may follow the key and b) the probability of a bigram (key + such tag). Then we use a script that runs through the corpus and looks for ambiguous words. For each such word the script checks whether the previous word is not ambiguous, and if so the script would consult the bigram model, choose the most probable tag out of the given and delete all the redundant analyses from the current word interpretation.

Let us illustrate how the model works. For example, the model contains a frequent nonambiguous bigram *N,m,pl* followed by *V,pres,pl,1*. When the script meets the tag *N,m,pl* followed by an ambiguous word with tags *V,pres,pl,1_V,pres,pl,3_V,inf,* it would keep *V,pres,pl,1* and delete the others.

This algorithm does not use any other statistics about contexts in which particular word analyses can be met, so its results can be erroneous. Surprisingly, the accuracy of this method was rather sufficient, and we will demonstrate it further.

We applied the algorithm to the testing corpus and received the following results for the ambiguity parameters and the effectiveness of the algorithm (Table 6):

Table 6. Ambiguity parameters of the testing corpus after applying the bigrams algorithm and the effectiveness of the algorithm

Corpus	Number of tokens	Percentage of ambiguous words	Ambiguity rate	Precision (%)	Recall (%)	F_1-score
Greek	866,091	38%	1.59	83	8.1	14.82
Yiddish	4,144,524	24.4%	1.65	78	—	—

As we can see from the table, this simplified and easy to execute model, surprisingly, demonstrates the same level of effectiveness as the intelligent data-driven Brill algorithm and even has the higher level of precision.

4.3. An interface for disambiguation by hand

All the approaches considered above are more or less effective, but all of them make mistakes. Every method considered earlier generated incorrect changes of tags, so that the correct tag for a particular word was deleted. Manual disambiguation is usually more accurate, but it is a very tedious process.

Imagine that a corpus has 3,000 instances of a bigram *ART, m, sg + N, m, sg_N, m, pl*, where the second word is ambiguous and has two possible tags. The correct tag is obvious, but a human would have to disambiguate this one simple bigram 3,000 times. It would be more convenient to resolve such morphological ambiguity just once and then automatically apply the result to all corresponding cases. For such disambiguation process we designed a program which interacts with a linguist and works as an automatic text processor.

The program collects ambiguous unigrams, bigrams and trigrams in the corpus and sorts them by frequency. Then the program shows one of the collected items to the user and offers to choose which variant is correct. The user can mark the correct answer or delete the wrong variants. If the user is not sure how to resolve ambiguities, they can be skipped. The accuracy of this method depends on the knowledge of language. Assuming that the linguist knows the language, this algorithm is very accurate.

This method stands closer to the rule-based methods as it does not depend on the data—the user can choose the right variant even when all the words in a bigram or trigram are ambiguous. Therefore, this method can be the first step in the process of disambiguation because its results cannot significantly change with the increase of unambiguous words in the corpus. In contrast, this method can supply data-driven methods with the higher number of unambiguous contexts and consequently improve their precision and recall while itself demonstrating supposingly high precision.

We received the following results for this user-guided disambiguation (Table 7):

Table 7. Ambiguity parameters of the testing corpus after applying user-guided disambiguation and the effectiveness measures for the algorithm

Corpus	Number of tokens	Percentage of ambiguous words	Ambiguity rate	Precision (%)	Recall (%)	F_1-score
Greek	866,091	31.8	1.50	84.8	31.1	45.51
Yiddish	4,144,524	34.3	1.88	97.5	—	—

As we can see, this method turned out to be very effective as it demonstrated both high recall and high precision. Actually, the value of precision is not equal to 100% because in some cases incorrect tags were deleted, but the word still had more than one analysis, which was counted as an imprecise case. In fact, this method did not generate incorrect tag changes, in contrast to the previous data-driven methods.

5. Results and discussion

We gave an overview of our methods and their effectiveness when they were applied to the training corpus. However, as it was mentioned earlier, the disambiguation becomes more accurate and extensive when several methods are combined.

The key point of the combination of methods is that data-driven methods work better if they are given more positive material. The more unambiguous data we have the more precise the method is and the higher the recall percentage is. Therefore, our aim is to use the methods which are not data-driven first, then to use methods which are more precise so that they could create more positive data for methods which are less precise.

All in all, we chose the following order:

1. user-guided disambiguation, which has high precision and is not data-driven;
2. the bigrams algorithm, which is data-driven, but with high precision;
3. the Brill algorithm (full disambiguation, not only by POS-tags), which has lesser precision;
4. rule-based disambiguator for ambiguity types left (for Greek).

Table 8 demonstrates the changes in process when we applied to the training corpus our disambiguation methods in this order:

Table 8. Ambiguity parameters and effectiveness measures for the combinations of methods

Corpus	Method	Percentage of ambiguous words	Ambiguity rate	Precision (%)	Recall (%)
Greek	user-guided	31.8%	1.50	84.80	31.10
	+ bigrams	29.0%	1.45	85.00	36.00
	+ Brill	26.0%	1.40	79.92	43.48
	+ rules	23.0%	1.35	82.41	50.60
Yiddish	user-guided	34.3%	1.89	97.50	—
	+bigrams	17.8%	1.48	82.70	—
	+ Brill	15.2%	1.39	81.70	—

As we can see from this table, the ambiguity rate gradually falls down with every method, and recall rises while precision stays on the high level. All this shows that every method indeed becomes effective if applied in the right combination with other methods.

In this paper, we have considered different disambiguation methods for the case when machine learning and supervised methods based on the pre-disambiguated corpus are not accessible to the researcher. We adapted several data-driven approaches such as the Brill algorithm and the Viterbi algorithm so that they became useful in our situation. Also we designed several techniques of our own such as user-guided disambiguation by bigrams and trigrams and supported our scheme with a conventional rule-based parser. In the end, we managed to resolve a significant number of ambiguous analyses in our corpora and proved that it could be done without using specific linguistic resources, such as training corpora disambiguated by hand.

Of course, it can be argued that the precision and recall we managed to achieve are not high enough to suit the needs of linguistic research. However, in the situation of the total absence of disambiguation tools for these languages developing approaches to disambiguation is vital, and, as the research concerning Swedish, Turkish and Icelandic shows, the quality of disambiguation can be improved, so we plan to adapt the solutions proposed for other languages with respect to disambiguation.

References

1. *Arkhangelskiy, T., Belyaev, O., & Vydrin, A.* (2012). The creation of large-scaled annotated corpora of minority languages using UniParser and the EANC platform. Proceedings of COLING 2012: Posters. Mumbai: The COLING 2012 Organizing Committee, 2012. Ch. 9. P. 83–91.
2. *Brill E.* (1995a). Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. Computational linguistics, 1995, V. 21, № 4, pp. 543–565.
3. *Brill E.* (1995b). Unsupervised learning of disambiguation rules for part of speech tagging. Proceedings of the third workshop on very large corpora. Somerset, New Jersey: Association for Computational Linguistics, 1995, V. 30, pp. 1–13.
4. *Goutsos, D.* (2010). The corpus of Greek texts: a reference corpus for Modern Greek. Corpora, 5 (1), 29–44.
5. *Van Halteren H., Daelemans W., Zavrel J.* (2001). Improving accuracy in word class tagging through the combination of machine learning systems. Computational linguistics, 2001, V. 27, № 2, pp. 199–229.
6. *Hatzigeorgiu, N., Gavrilidou, M., Piperidis, S., Carayannis, G., Papakostopoulou, A., Helgadóttir, S.* (2004). Testing data-driven learning algorithms for pos tagging of icelandic. Nordisk sprogteknologi, 2000–2004.
7. *Kermanidis, K. L., Fakotakis, N., & Kokkinakis, G.* (2002). DELOS: An Automatically Tagged Economic Corpus for Modern Greek. In Proceeding of LREC 2002. P. 718–722.
8. *Kupiec J.* (1992). Robust part-of-speech tagging using a hidden Markov model. Computer Speech & Language, 1992, V. 6, № 3, pp. 225–242.
9. *Marier, F., & Sjödin, B.* (2003). A part-of-speech tagger for Swedish using the Brill transformation-based learning. Projektarbeten 2003, 102.
10. *Sak, H., Güngör, T., & Saraçlar, M.* (2007). Morphological disambiguation of Turkish text with perceptron algorithm. In Computational Linguistics and Intelligent Text Processing (pp. 107–118). Springer Berlin Heidelberg.
11. *Spiliotopoulou, A., & Demiros, I.* (2000). Design and Implementation of the Online ILSP Greek Corpus. In Proceeding of LREC 2000.