"God Wat Þæt Ic Eom God" — An Exploratory Investigation Into Word Sense Disambiguation in Old English

Martin Wunderlich
Center for Information and Language Processing (CIS)
University of Munich (LMU)

Alexander Fraser
Center for Information and Language Processing (CIS)
University of Munich (LMU)

P. S. Langeslag*
Dept. of English Philology
Medieval English Language and Literature Section
University of Göttingen

Abstract

Natural Language Processing (NLP) of historical languages is an understudied area. Much previous work has focused on the problems of normalization and POS tagging. In contrast, we consider a new problem, word sense disambiguation (WSD). We provide a survey of previous work on processing of historical languages and discuss what we can and cannot apply to the problem of WSD, specifically WSD of Old English (OE). We then annotate a new resource for supervised WSD, which we make available. Finally, we carry out proof-of-concept experiments, followed by a discussion of several promising areas for future work.

1 Introduction

We consider the important task of Word Sense Disambiguation (WSD) for historical languages, which to the best of our knowledge has not been studied extensively yet. WSD is at the heart of many applications in Natural Language Processing (NLP). For instance, in order to correctly translate polysemous or homonymous words in a machine translation system, one needs to disambiguate different word senses. The target language might make a clear lexical distinction where the words in the source language are homographs (Yarowsky, 2010). Examples of polysemy/homonymy would be lexical items, such as "Python" (the snake or the programming language?) in written language, the homophones "Perl" vs. "Pearl" in speech, or — as an example for a homograph — the word "God" in the (synthesized) Old English phrase "god wat þæt ic eom god" ("God knows I'm good").[1]

*Email addresses: martin.wunderlich@campus.lmu.de, fraser@cis.lmu.de, ps@langeslag.org

[1]Note that the latter ambiguity could easily be resolved, if there were a POS tagger for Old English.

We consider WSD for a language that is particularly difficult in this regard but also very interesting. Old English is a Germanic language that was spoken on the British Isles approximately from 450 to 1150 AD, then it gradually transformed into Middle English (ME), particularly under the influence of the conquerors' languages Old Norse and (Norman) French. In the area of NLP, Old English is an under-researched language. It has received relatively little attention — unlike its contemporary variant — and there are few digital resources and tools.

Our contributions are: First, we present a brief survey of the NLP literature on historical languages. Second, we annotate new gold standard training and testing data and make it available for future use. Third, results from a proof-of-concept study are used to show how the problem of WSD for OE can be concretely approached.

The initial results are promising, even with basic techniques. This demonstrates the feasibility of WSD and might motivate work on expanding the inventory of data and NLP tools available for OE, such as POS taggers and stopword lists, which can be applied in WSD and other tasks.

The remainder of this work is structured as follows: In section 2 an overview of the history of Old English is given. In section 3 we present a survey of works on the application of NLP to historical languages. Section 4 briefly summarizes existing work on WSD and details the methods that are being used in the present work. The focus of section 5 is the description of the practical development work that was carried out as part of this project. This section also includes an overview of existing digital resources for NLP as applied to OE, covering both digital corpora/lexica and existing tools. The steps taken for preparing and preprocessing of the digital resources are also described in full detail. Section 6 presents the proof-of-concept evaluation. Finally, section 7

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology, pages 39–48, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

summarizes our findings and discusses several avenues of future work.

2 An Extremely Brief History of Old English

From about 800 BC, the British Isles were populated by Celtic settlements.[2] After initial Roman military expeditions, starting with Julius Caesar in 55 BC, the Roman province of Britannia was established by the year 43 AD. In the 5th century AD, however, the Roman heartland came under pressure and so the troops were withdrawn. Their withdrawal was complete by 410 AD.

This vacuum of power was used in the 5th century by tribes from the north (the Scots and Picts) to push into the southern part of the British Isles, while at the same time Germanic tribes from the European mainland — the Angles, Saxons, and Jutes — likewise made their way to what came to be known as England. The Saxons settled in the south, whereas the Angles settled in the north and the Jutes in Kent.[3] The Anglo-Saxons quickly established rule over Britain and by the end of the 5th century Saxons and Celts lived under the "Rex Anglorum".

The Anglo-Saxons had brought their culture and languages with them, which were quickly adapted and transformed by the local population, giving rise to what is known as the Old English language. This Germanic language retained many grammatical features of its parent languages, which makes it quite distinct from contemporary English. The Old English alphabet was based on Latin and consists of 24 letters:

a **æ** b c d **ð** e f **ȝ**/g h i l m n o p r s/**ſ** t **þ** u **ƿ**/w x y

The language has a case system with five cases (nominative, genitive, dative, accusative, and vestiges of an instrumental) and three numbers (singular, dual and plural). OE is a strongly inflected language, as can be seen from the following two examples:[4] 1) **se** guma geseah **þā** cwēn ("the man saw the woman"); 2) **sēo** cwēn geseah **þone** guma**n** ("the woman saw the man")

The variations for case, gender, and number are clearly visible here in the definite article (highlighted). Also note the suffix for "guma" when used as a direct object in the second sentence (and the lack of such an inflection for "cwēn").

Irish and Roman missionaries introduced the Latin language at large scale (which had left few traces during the previous Roman occupation). Several word borrowings can be traced to Latin roots, such as "missa" – "mæsse" ("Mass"), "presbyter" – "prēost" ("priest") and "calendae" – "calend" ("calendar"). The OE language was further influenced by Old Norse, following several waves of Scandinavian raids and invasions first recorded in 787 and recurring into the late eleventh century. After the Treaty of Wedmore in 886, an area known as the "Danelaw" was established in northeastern England. Names for locations and people can be traced to these Scandinavian roots, such as "Whenby" or "Skewsby", "Jackson" or "Davidson". The initial "sk" in words such as "skirt", "skin" or "skill" has Old Norse roots. Also, common words like "same" or "give", and even some closed-class pronouns can be traced back to Old Norse: The 3rd person plural forms of the pronoun have Scandinavian roots.

The entire OE corpus that survives consists of only approximately 24,000 word types, around 15% of which have remained in Modern English and 3% are loan words (Crystal, 2010, page 27). Most of this corpus is in the West Saxon dialect, since under King Alfred's rule many works were translated from Latin into OE. The two other main dialects are Northumbrian and Mercian. A lack of standardized orthography, combined with sound changes, morphological, and dialectal variations, acted increase the number of word types and gave rise to word variations.[5]

3 Related Work On Historical Languages

As pointed out in the introduction, OE is an under-resourced and under-researched language when it comes to the field of NLP. Nevertheless, a few related studies that cover OE and other historical languages can be found. Sukhareva and Chiarcos (2014) examine the possibility of using data from related languages and dialects to compensate for the sparseness of annotated corpus data in OE and other historical languages. They use parallel biblical texts to train a dependency

[2] The following section is largely based on Crystal (2010, pages 6-29) and Schirmer and Esch (1977, pages 2-20)

[3] At least according to the traditional (and probably simplified) account by the Northumbrian monk Bede; cf. (Crystal, 2010, page 6)

[4] Taken from Crystal (2010).

[5] Such as "wunderlic", "wundarlic", "wundorlic", which might be translated as "peculiar", "strange".

parser and find that annotation projections derived from word alignments allow for cross-language parser adaptation. The authors speculate "[...] that languages separated for 1000 years (OE-IS) or more are too remote from each other to provide helpful background information, but that languages separated within the last 750 years (ME-DE) or less are still sufficiently close."[6] (Sukhareva and Chiarcos, 2014, page 15) In the context of the present work this means that in terms of the temporal distance resources for ME might be useful, but in terms of the relatedness of OE and ME, the languages might be too different, due to the influences described in the previous section.

Pennacchiotti and Zanzotto (2008) evaluate to what extent existing NLP tools for contemporary Italian are suitable for POS tagging applied to fourteenth-century Italian using a corpus of fourteen major Italian literary works, such as Dante Alighieri's *Divina Commedia* from 1321. In addition, the authors test in what manner simple modifications and customizations of the existing tools might improve their application to late medieval Italian. The evaluated accuracy of the POS tagging ranges between 0.54 and 0.90. In conclusion, the authors find that the results "[...] support our initial claim that the dictionary and the Chaos parser for contemporary Italian are insufficient for the analysis of ancient texts, as there exists a significant gap in dictionary coverage between contemporary and ancient texts." (Pennacchiotti and Zanzotto, 2008, page 378) The authors also propose possible improvements, such as manually building a lexicon for each period, leveraging manually annotated corpora or adapting existing models by applying rules to capture morphological variations.

In a similar fashion, Meyer (2011) uses existing NLP resources for contemporary Russian to tag Old East Slavonic texts, by first annotating the modern version and then projecting part of the annotation back onto the corresponding original forms, based on a parallel corpus consisting of old and modern versions of the same texts. Meyer presents a system that goes through steps of sentence alignment, "guessing" of morphological categories, word alignment, creation of hyperlemmata[7] and, finally, annotation projection. The main result of this work is the finding that this method can be used to successfully derive morphosyntactic annotations in a process that is based on the disambiguation of the output of a morphological guesser with the help of aligned Modern Russian word forms and associated tags.

Bollmann (2013) carried out similar work in the area of POS tagging on historical German texts from two corpora: the 15th century Anselm corpus and GerManC-GS with texts from the 17th and 18th centuries. Various steps of normalization and different parametrizations are derived automatically by the Norma tool (Bollmann et al., 2012). POS tagging on the historical texts is evaluated in three different scenarios: first, tagging on the simplified, but otherwise unmodified, original texts; second, tagging on the gold-standard normalizations; and third, tagging on texts which have been normalized automatically. The author reports accuracy results of around 69.6% for Early Modern German texts and POS tagging results of 81.92% for the historical texts when tagging on gold-standard normalizations (vs. 95.74% for modern data) (Bollmann, 2013, page 16).

In a study pertaining to Middle English (Moon and Baldridge, 2007), tags from present day English source texts were projected to Middle English texts using alignments from a parallel Biblical text. The authors report a "[...] tagging accuracy in the low 80's on Biblical test material and in the 60's on other Middle English material." (Moon and Baldridge, 2007, page 390). This work was based on the annotated Penn-Helsinki Parsed Corpus of Middle English, containing texts from from around 1150 to 1500. This corpus contains approximately 1,150,000 words of running text from 55 sources. The texts are provided in three forms: raw, tagged, and parsed. Using a bigram tagger, "[r]esults were improved further by training a more powerful maximum entropy tagger on the predictions of the bootstrapped bigram tagger, and [the authors] observed a further, small boost by using Modern English tagged material in addition to the projected tags when training the maximum entropy tagger" (Moon and Baldridge, 2007, page 398).

As regards Early Modern English, Baron and Rayson (2008) carried out experiments using

[6]The language codes here stand for: Old English (OE), Middle English (ME), Middle Icelandic (IS), and Early Modern High German (DE).

[7]That is, "[...] an artificial label bundling together corresponding lemmata of different diachronic stages." (Meyer, 2011, page 274)

automatic spelling normalization. In the process of this, a tool was created called VARD 2, which could possibly be adapted for OE. Normalizing the spelling across the corpus helps to reduce the spelling variations that derive from the non-standardized orthography and thus reduce the noise that stems from these variations.

As is evident from the works cited above, the primary focus of NLP on historical texts has been the problem of POS tagging and the possibility of applying existing tools for contemporary languages to their historical counter-parts. Detailed studies on WSD for historical languages, particularly on OE texts, seem to be non-existent. Also, the works quoted above focus mainly on annotation projection, an approach which is not applicable for the WSD task since no sense-annotated corpus of a sufficiently closely related language exists to the best of our knowledge. The existing body of work shows that standard classification methods, such as maximum entropy, can be used successfully and that parallel corpora are a useful resource for historical languages, but only if the two languages are sufficiently closely related. For Old English, however, no such parallel corpus exists, so the present work is based on a monolingual body of text.

4 Methodological Background on WSD and Machine Learning Techniques

If the meaning of word is its usage in the language, as Wittgenstein claimed,[8] then it should be possible to derive the meaning by closely examining this usage. One aspect of the usage is the context in which a word appears with a certain meaning or word sense and, consequently, techniques for Word Sense Disambiguation focus on the context of a word to select the most likely word sense from a given "sense inventory" (Yarowsky, 2010). Essentially, WSD is a classification task using the context words in the sentence or paragraph and, possibly, additional information such as their POS tags, as evidence (Yarowsky, 2010). The term "sense inventories" here can mean any form of dictionary-based repository that maps lemmas or lexical items to word senses. The task of WSD consists of a semantic analysis or interpretation with the goal of deriving the meaning of an utterance. In WSD each word can be considered a classification problem in its own right (Cabezas et al., 2001), for the purpose of which each word instance is represented as a collection of feature-value pairs in vector form and the correct category assigned to this training instance in form of a unique sense ID or label.

Supervised machine-learning algorithms can be applied to this WSD classification problem. The ability to distinguish different word senses is "learned" from sense-labeled training examples of polysemous/homonymous words in the context of a sentence or paragraph. The context could, for instance, be a window size of 50 words to the left or right of the target word, which is cited by Yarowsky (2010) as a typical window. This window is then be converted to a bag-of-words feature vector, with either binary values, signifying the presence or absence, or using a more fine-grained representation, such as TF-IDF. Other features, such as the POS of a context word at a given position relative to the target word, might also be used (Yarowsky, 2010). Since an ambiguous word might have more than two meanings, the task can be modeled as a multi-class classification or a binary ("one versus all") classification. In the present work, one of the findings has been that binary classification in general performs better than multi-class classification.

Stevenson differentiates four categories for WSD tasks (Stevenson, 2003): 1) Semantic disambiguation where there are no restrictions as to the number or kinds of senses.[9] 2) Semantic tagging: Also known as the "all-words task" whereby all words have to be annotated with a specific word sense. 3) Sense disambiguation whereby some words (not all) are to be tagged with a specific sense from a lexicographical resource. 4) Sense tagging, whereby all words are to be tagged with lexical senses.

The task in the present work would fall into the third category, since only a selection of polysemous words is being tagged with word sense classes from a lexicon.

When running any kind of machine learning algorithm, it is useful to have a baseline that the results can be evaluated against. Stevenson presents a number of possible baselines in WSD tasks (Stevenson, 2003). However, for our purposes only two of these are relevant: 1) the random selection of a word sense and 2) the selection of

[8] Ludwig Wittgenstein: *Philosophische Untersuchungen*, §43, page 40. Suhrkamp Verlag, Frankfurt a.M., 6th ed., 2013.

[9] Also referred to as "word sense discrimination".

the most frequent word sense (from the training set). The other three baseline metrics proposed by Stevenson build on the Lesk algorithm,[10] which uses lexical overlap between the target word's context and dictionary definitions for classification and is therefore not applicable in our case, since the corpora here are in a language (Old English) that is different from the lexicographic definitions (Modern English). Usually, the Lesk algorithm should be strongly considered as a WSD algorithm and might be reconsidered for the work presented here if or when dictionaries with definitions in OE become available in the future. In terms of classification methods, the present work compares Naïve Bayes with Maximum Entropy, both evaluated against random and most frequent baselines.

5 Old English NLP Resources Used in the Present Work - Selection, Preparation, and Preprocessing

In the following section, the digital resources that formed the basis of our work are described. Statistics on the Old English corpus and lexicon are presented in sub-sections 5.1 and 5.2. The third sub-section (5.3) provides a brief overview of the preparation of the data used for training the machine-learning algorithm. Also, the feature extraction steps that were employed to generate feature vectors from the corpus data are described in that section.

5.1 The "Dictionary of Old English Corpus" (DOE Corpus) and Preprocessing Applied To It

Old English corpora are not as abundant as their contemporary counter-parts, but nevertheless some specimen can be found. In this present work, one main corpus was used, the DOE Corpus[11] or, more accurately, "The Dictionary of Old English Web Corpus", compiled by Antonette di Paolo Healey with John Price Wilkin and Xin Xiang (diPaolo Healey et al., 2009)[12]. The DOE corpus contains the text of at least one manuscript witness for every extant Old English text, including both prose and verse, as well as glosses, glossaries, and inscriptions

A number of preprocessing steps were undertaken, such as tokenization on sentence and word level. Other potentially useful pre-processing steps, such as lemmatization and POS tagging, were not possible, due to the lack of existing tools, but future work might use POS tagged data from the YCOE corpus. The following table 1 lists statistical information on the corpus:[13]

Number of HTML documents	3,037
Token count	3,786,753
Type count	343,135
Ratio of (token count / type count)	ca. 11
Total number of sentences	234113
Average sentence length	5.5
Minimum sentence length	1
Maximum sentence length	263

Table 1: Corpus statistics for the DOE corpus

5.2 Lexicographic Resources

In order to obtain a set of polysemous words, the Dictionary of Old English (DOE)[14] was employed. The DOE provides vocabulary from the first six centuries (600 - 1150 AD) of the English language and list entries for approx. 12,500 terms, currently ranging from letters A through G. The DOE comes in the form of HTML documents. The word counts by initial letter are given in table 2 (with some minor word count differences between the counts on the DOE website and the actual counts in the corpus).

The HTML format was parsed into a Java class structure and from this structure polysemous candidate terms were extracted. The criteria for the extraction were as follows:

- minimum token count 200
- minimum word length 3 characters

[10] For a detailed description of the Lesk algorithm see, for instance, Jurafsky and Martin (2008, pages 680f)

[11] Downloaded from the University of Oxford Text Archive - http://ota.ox.ac.uk; last accessed 2014-12-25

[12] An alternative might have been the York-Toronto-Helsinki Parsed Corpus of Old English Prose (YCOE), a 1.5 million word syntactically-annotated corpus of Old English prose texts (for which a corpus reader is included in NLTK), but the DOE corpus was chosen due to the larger volume. YCOE is a subset of DOE. For details see http://www-users.york.ac.uk/~lang22/YCOE/YcoeHome.htm

[13] The difference between the type count here and the OE type count of 24,000 provided earlier derives from the absence of any normalization and lemmatization in our software. Types are the raw word types exactly as they appear in the DOE corpus. So, for instance, "Fæder" and "fæder" would be counted as two different word types. The motivation for this is that we wanted to provide the generic text data whereas normalization would have lead to a loss of information.

[14] The DOE resources were last accessed and downloaded 2014-12-25 under http://tapor.library.utoronto.ca/doe/andhttp://tapor.library.utoronto.ca/doecorpus/

Letter	Counts on DOE website	Counts in HTML files
A	1,539	1,540
Æ	623	623
B	2,264	2,285
C	1,409	1,418
D	921	927
E	1,480	1,481
F	3,013	3,029
G	1,319	1,322

Table 2: Word counts from DOE

- non-Latin (i.e. no “dictum”, “confundantur”, “magister”...)
- minimum number of dictionary entries 2 (obviously)
- common nouns
- no proper nouns (e.g. no “Egypta”, “Micel”, “Iulianus”...)

The candidates were then reviewed manually to obtain an initial list of ten polysemous terms with sufficiently diverging word senses, as checked against the DOE definitions. From this shortlist of ten terms, we excluded those where the distribution of word senses in the randomly selected concordance matches was too skewed.[15] Table 3 in the appendix gives an overview of the seven remaining terms with their sense labels and definitions.

For the remainder of this present work, we will be focusing on the WSD results for the term “boc” as a representative and sufficiently ambiguous term.[16] Two examples for concordances of the target term “boc” shall serve to illustrate the format of the data (with doc ID and line ID for the DOE corpus):

- Doc ID: ÆGenPref; Line ID: 003800 (117); Ic bidde nu on Godes naman, gyf hwa ðas **boc** awritan wille, ðæt he hi gerihte wel be ðære bysne, for ðan ðe ic nah geweald, ðeah ðe hi hwa <to> woge gebringe ðurh lease writeras, & hit bið ðonne his pleoh na min: micel yfel deð se unwritere, gyf he nele his gewrit gerihtan.[17]
- Doc ID: MtGl (Li); Line ID: 062600 (19.7); dicunt illi quid ergo moses mandauit dari librum repudii et dimittere cuoedon him huæt forðon bebead sella **boc** freodomas & forleta[18].

One major drawback of the DOE is that it seems to have been engineered for use by human scholars and not by machines. There is no downloadable version in (TEI-)XML and there is no API for convenient access by other systems. Therefore, the dictionary had to be processed in HTML form and the information needed to be extracted from raw HTML tables into a structured Java object format.

To generate the training data, 100 concordance sentences for each word were randomly selected from the DOE corpus. Each occurrence was manually labeled with the sense ID of the top-level sense as per the DOE definition. These annotations were then verified by a second annotator. The final distribution is shown in figure 1. For the target word “boc”, the two instances of sense class C were removed. Also, one instance which used the word in the sense of the tree “beech” was removed and two instances could not be classified with sufficient reliability. This left a total of 95 training instances (with the three labels A: 33; B: 20; D: 42).

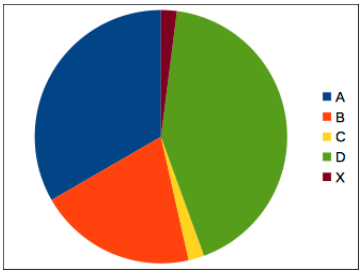

Figure 1: Word sense distribution of “boc”

The effort of the manual annotations was quite considerable, taking at least an hour per word, not including the quality checks and reviews. For a more comprehensive study, it would be possible perhaps to extract sense-labeled training data from the DOE files directly. This, however, could lead to sparsity issues, since on the lowest level

[15] This which would have lead to sparsity problems. The excluded terms with their word sense distribution were: “andlang” (1: 0; 2: 0; A: 16; B: 0); “ban” (A: 88; B: 6; X:6); “eadigen” (1: 21; 2: 4). The numbers do not add up 100 because the labeling was canceled once the skewed distribution became obvious. Sense labels are those from the DOE definitions.

[16] The additional data for the other target terms are available via the following URL (together with links to the code repository): `http://www.cis.uni-muenchen.de/~martinw/`

[17] Translation: “I ask now in the name of God, if anyone desires to copy this book, that he corrects it well by the exemplar, because I have no control if someone brings it to error through lesser scribes, and it is then his peril, not mine. The bad scribe does much evil if he will not correct his errors.” - translated by Brandon W. Hawk: `http://brandonwhawk.net/2014/07/30/aelfrics-preface-to-genesis-a-translation/` - last accessed 2015-05-14.

[18] Note the mixed language example with both Latin and Old English in this second example. The OE here is a gloss to the Latin text (Matthew 19:7) from the Vulgata. In the King James Bible, this verse is translated as: “They say unto him, Why did Moses then command to give a writing of divorcement, and to put her away?”

each meaning definition might have only two or three example sentences. It would be possible to circumvent this problem by merging all sub-level meanings into the top-level, but this procedure would have to be carefully evaluated first for its validity. A threshold for the minimum number of sample sentences could also be introduced, but again this might create sparsity problems.[19]

5.3 From Corpus to Feature Vectors

In order to train the various learning algorithms, the training data needs to be converted to an abstracted representation in the form of feature vectors, one vector per instance of a training word. A feature here is a particular characteristic derived from the context of the target word and the feature vector is a collection of several such features. Ng lists a number of possible feature types (Ng and Zelle, 1997): 1) surrounding words (unordered set within fixed size window or word from the entire sentence); 2) local collocations (short sequence with word order); 3) syntactic relations (e.g. verb-object relations); 4) POS of context words and morphological features.

Cabezas et al. (2001) distinguish between two types of features 1) feature f_{WIDE} will be non-zero, if f appears in wide context of target word w; 2) feature $f_{COLL(x,w)}$ will be the token $\pm$ x positions to the right or left of w.

The full feature set F in this case would then be the union of $f_{WIDE} \cup f_{COLL}$. Following these authors, two initial types of feature vector were obtained: 1) Unordered BoW vector, which comprised all words in the same paragraph as the given target word within a token window of $\pm$ x tokens, where x was varied between 1 and 20. 2) Collocation vector, by creating features for ordered words in a window of $\pm$ 20 words on either side of the target word. The following is an example for such a feature vector (bag-of-words) for a window size of n=5 for the example sentence from section 5.2:

```
godes(9)=1.0
naman(10)=1.0
gyf(11)=1.0
hwa(12)=1.0
ðas(13)=1.0
awritan(14)=1.0
wille(15)=1.0
ðæt(16)=1.0
he(17)=1.0
hi(18)=1.0
```

[20]

6 Evaluation Metrics, Experiments, and Results

6.1 Evaluation Metrics

The assessment of the various classification methods requires solid and pre-defined evaluation metrics. These metrics should then be compared to pre-defined upper and lower bounds, for instance those given by Gale, who lists 75% lower bound and 96.8% upper bound, derived from the agreement of human judges (Gale et al., 1992). During the test runs for each trained classifier the following metrics were calculated for each classification (per target word and per one-vs-all classification as regards the word senses): accuracy,[21] precision, recall, and balanced F1 measure.

6.2 Experiments and Results

We compared different machine-learning techniques for the use of Old English WSD, using two classifier types: Naïve Bayes and Maximum Entropy. Both were provided by the MALLET machine-learning library written in Java (McCallum, 2002).

For each type of learning algorithm, a multi-class classification was compared to the binary classification of creating one-vs-all classifiers per sense class. As the baseline to compare the results against, a random selection and a "most common sense" heuristic were both used.[22]

The two classification algorithms were trained on feature vectors as follows: 1) BoW vector with a token window between 1 and 20 tokens. 2) Collocational vector (i.e. including positional information) with a token window between 1 and 20 tokens. In the appendix, figure 4 presents the results from the baseline classification. Figure 5 presents the results from actual classification for

[19] A different approach that does not rely on hand-labeled data would be the use of clustering techniques, such as graph-based methods or using lexical expansion, to generate sense clusters in an unsupervised manner, as described e.g. in Bordag (2006), Biemann (2012) or Miller et al. (2012). This unsupervised approach is known as *Word Sense Induction* and might be applied to OE in future work.

[20] Adapted from the output of MALLET's PrintInputAndTarget pipeline step.

[21] Also known as the "exact match criterion" (Stevenson, 2003)

[22] Since these baseline classifiers did not exist in MALLET, they were created from scratch as part of this present work and have been accepted into the project as a contribution via GitHub. Accepted on 2015-01-19, see https://github.com/mimno/Mallet/

target term "boc" using Naïve Bayes and Maximum Entropy.

It can be seen from this latter table that the best results in term of classification accuracy for the target term "boc" were achieved for a Naïve Bayes classifier using a bag-of-words model and a binary classification task ("one-vs-all") for sense ID "D". The same combination also gave the best values for precision (0.85), recall (0.83), and F1 (0.82). Overall, from the two types of classification methods, Maximum Entropy yielded a slightly better average accuracy of 0.734 (as compared to Naïve Bayes with 0.729). Naïve Bayes scored slightly higher in terms of overall average F1 measure with 0.666 (MaxEnt: 0.658), but the differences are probably negligable.

7 Conclusion and Potential Future Work

This present work has tried to demonstrate in which manner modern methods of statistical text processing can be used for the purposes of word sense disambiguation on an under-resourced language like Old English, provided that corpora and dictionary resources exist in digital form.

In the future, more tools for processing Old English texts might become available, such as POS taggers and NE extractors, which could be used to generate richer feature vectors.[23] Also, such tools would be useful in the preprocessing steps and could reduce words to their lemmas, which might help improve classification results. The features provided by the different window sizes could be analyzed closely for sparsity issues and a form of count-based cutoff might be implemented to try to be more robust. Other methods of dimensionality reduction, for instance knowledge-free stemming (Porter-stemming for OE, simple learned stemming, or simple truncation) could also be applied to reduce the sparsity of features. Also, it could be possible to use existing data of Modern English to train ML algorithms for WSD, although one might have reservations about the prospects, since OE and ModE are syntactically and lexically very different languages.[24]

In this work we focused on the use of Naïve Bayes and Maximum Entropy as classification methods. Other common machine-learning techniques that have been applied for WSD could also be used, such as Bayesian networks (Bruce, 1995), content vector models in combination with clustering techniques and Singular Value Decomposition (Schütze and Pedersen, 1995), (Schütze, 1998), or Artificial Neural Networks (Veronis and Ide, 1990).

Future work could also apply the classification methods in combination with bootstrapping techniques,[25] especially when the set of sense-labeled training data is relatively sparse (cf. Ng and Zelle (1997)). Since at present there is no single best WSD method, it might also make sense to combine several different classifiers in such a fashion, even in cases where there is a more satisfactory abundance of training data and combine these classifiers in a framework of several WSD sources and systems (Stevenson, 2003).

Acknowledgments

The authors would like to acknowledge the valuable help provided by Winfried Rudolf (University of Göttingen) who helped with translations and general comments. Further, we would like to thank the three anonymous reviewers whose comments led us to improve the paper.

References

A. Baron and P Rayson. 2008. Vard 2: A tool for dealing with spelling variation in historical corpora. In *Online Proceedings of the Aston Postgraduate Conference on Corpus Linguistics*, Birmingham, U.K.

Chris Biemann. 2012. Word Sense Induction and Disambiguation. In *Structure Discovery in Natural Language*, pages 145–155. Springer, Berlin, Heidelberg.

Marcel Bollmann, Stefanie Dipper, Julia Krasselt, and Florian Petran. 2012. Manual and semi-automatic normalization of historical spelling - case studies from early new high german. In *Proceedings of the*

[23] Alternatively, the syntactical and POS information provided by the YCOE corpus might be parsed and applied for WSD.

[24] As Moon et al. note on the difference between ME and ModE: "It is also questionable whether it would still be robust on texts predating Middle English, which might as well be written in a foreign language when compared to Modern English." (Moon and Baldridge, 2007, page 398)

[25] In bootstrapping the available training data is used as an initial seed set to train a classifier. The trained classifier is then applied to a larger corpus and the sense-labeled word instances gained from this are added to the training set. A threshold should be set in advance so that only classifications with a certain confidence score get added. Bootstrapping can also be used to train a second classifier on the results of a first one, a form of multi-engine WSD, as used for instance in the system by Stevenson (2003).

11th Conference on Natural Language Processing (KONVENS 2012), LThist 2012 workshop, pages 342–350.

Marcel Bollmann. 2013. Pos tagging for historical texts with sparse training data. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability in Discourse*, pages 11–18, Sofia, Bulgaria, August. Association for Computational Linguistics.

Stefan Bordag. 2006. Word Sense Induction: Triplet-Based Clustering and Automatic Evaluation. In *EACL*. EACL.

Rebecca Bruce. 1995. A statistical method for word-sense disambiguation (phd thesis).

Clara Cabezas, Philip Resnik, and Jessica Stevens. 2001. Supervised sense tagging using support vector machines. In *The Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems*, SENSEVAL '01, pages 59–62, Stroudsburg, PA, USA. Association for Computational Linguistics.

D. Crystal. 2010. *The Cambridge Encyclopedia of Language*. The Cambridge Encyclopedia of Language. Cambridge University Press.

Antonette diPaolo Healey, John Price Wilkin, and Xin Xiang, editors. 2009. *Dictionary of Old English Web Corpus*. Dictionary of Old English Project.

William Gale, Kenneth Ward Church, and David Yarowsky. 1992. Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In *Proceedings of the 30th Annual Meeting on Association for Computational Linguistics*, ACL '92, pages 249–256, Stroudsburg, PA, USA. Association for Computational Linguistics.

Daniel Jurafsky and James H. Martin. 2008. *Speech and Language Processing (2nd Edition) (Prentice Hall Series in Artificial Intelligence)*. Prentice Hall, 2 edition.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Roland Meyer. 2011. New wine in old wineskins? - tagging Old Russian via annotation projection from modern translations. *Russian Linguistics*, 35(2):267–281.

Tristan Miller, Chris Biemann, Torsten Zesch, and Iryna Gurevych. 2012. Using Distributional Similarity for Lexical Expansion in Knowledge-based Word Sense Disambiguation. In *COLING*, pages 1781–1796.

Taesun Moon and Jason Baldridge. 2007. Part-of-speech tagging for middle English through alignment and projection of parallel diachronic texts. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 390–399, Prague, Czech Republic, June. Association for Computational Linguistics.

Hwee Tou Ng and John M. Zelle. 1997. Corpus-based approaches to semantic interpretation in NLP. *AI Magazine*, 18(4):45–64.

Marco Pennacchiotti and Fabio Massimo Zanzotto, 2008. *Natural Language Processing across time: an empirical investigation on Italian*, volume 5221, pages 371–382. Springer.

Walter F. Schirmer and Arno Esch. 1977. *Kurze Geschichte der englischen und amerikanischen Literatur*. Dtv ; 4291 : Wissenschaftliche Reihe. Dt. Taschenbuch-Verl., München, 4. edition.

Hinrich Schütze and Jan Pedersen. 1995. Information retrieval based on word senses. In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, USA.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–124.

Mark Stevenson. 2003. *Word sense disambiguation : the case for combinations of knowledge sources*. CSLI studies in computational linguistics. CSLI Publ., Stanford, Calif.

Maria Sukhareva and Christian Chiarcos. 2014. Diachronic proximity vs. data sparsity in cross-lingual parser projection. a case study on germanic. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, pages 11–20, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Jean Veronis and Nancy M. Ide. 1990. Word sense disambiguation with very large neural networks extracted from machine readable dictionaries. In *Proceedings of the 13th Conference on Computational Linguistics - Volume 2*, COLING '90, pages 389–394, Stroudsburg, PA, USA. Association for Computational Linguistics.

D. Yarowsky. 2010. Word sense disambiguation. In Alexander Clark, editor, *The handbook of computational linguistics and natural language processing*, Blackwell handbooks in linguistics. Wiley-Blackwell, Oxford, 1. publ. edition.

Appendix - Detailed Results

On the next page, we present detailed results for the term "boc" as discussed in the experimental section.

Target term	Token count	DOE definitions with IDs/labels
Anweald	242	A. power, sovereignty, sway B.1. a sovereign's or lord's dominion: realm, domain, empire; B.2. referring to the world considered as God's dominion: dominion C. the name of the sixth order of angels in the celestial hierarchy: Powers
Are	308	A. honour B. mercy, grace, favour, help C. property, possession(s), goods, resources
Boc	567	A. book B. major division of a larger work C. register, record, list D. legal document
Dryhten	261	1. in poetry and laws: lord, ruler, chief 2. the Lord, God the supreme ruler 3. lord, applied to a pagan god
Fæder	416	A. father (of humans) B. of supernatural beings / abstractions: father
For	955	1. action of going, state of movement, motion 2. journey, trip, voyage; fore geferan / gefremman - to go on / make a journey; 3. armed foray; march of an army 4. rendering accessus, here the approach, access 5. path, course; here figurative: way of life, course of conduct 6. glossing vehiculum means of transport, vehicle, conveyance
Fultum	574	1. help, aid, assistance, support, succour 2. concrete: someone who or something which provides help, support 2.a. supporter (of someone gen.; of a monastery, into and dat.) 2.b. referring to military support in the form of a force, troop, army 2.c. in medical recipes: a remedy

Table 3: APPENDIX — WSD target words from the DOE corpus with labeled definitions (sense labels are the original ones from the DOE).

Training algorithm	**Classification type**	**Vector type**	**Accuracy**		**Precision**		**Recall**		**F1**	
			Avg	Std Dev	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
rnd	A vs. not A	bow	0.55	0.13	0.49	0.27	0.51	0.30	0.47	0.26
rnd	A vs. not A	coll	0.57	0.14	0.53	0.31	0.57	0.28	**0.50**	0.25
rnd	B vs. not B	bow	**0.66**	0.13	0.55	0.36	0.56	0.36	0.49	0.33
rnd	B vs. not B	coll	0.64	0.17	0.51	0.41	**0.58**	0.39	0.45	**0.38**
rnd	D vs. not D	bow	0.49	**0.22**	0.52	0.28	0.49	*0.26*	0.48	0.23
rnd	D vs. not D	coll	0.53	0.17	0.53	*0.25*	0.53	*0.26*	**0.50**	0.21
rnd	multi	bow	0.38	0.18	*0.38*	0.33	0.37	0.33	0.32	0.29
rnd	multi	coll	0.37	0.13	0.41	0.35	0.41	0.36	0.32	0.28
mostfreq	A vs. not A	bow	0.35	0.16	0.68	0.35	0.50	**0.51**	0.25	0.28
mostfreq	A vs. not A	coll	0.34	0.14	0.67	0.35	0.50	**0.51**	0.25	0.27
mostfreq	B vs. not B	bow	*0.16*	*0.10*	0.58	**0.43**	0.50	**0.51**	*0.14*	*0.17*
mostfreq	B vs. not B	coll	0.17	0.12	0.59	**0.43**	0.50	**0.51**	*0.14*	0.19
mostfreq	D vs. not D	bow	0.42	0.18	0.71	0.32	0.50	**0.51**	0.29	0.31
mostfreq	D vs. not D	coll	0.50	0.14	0.75	0.27	0.50	**0.51**	0.32	0.34
mostfreq	multi	bow	0.37	0.19	**0.79**	0.32	0.43	0.50	0.27	0.36
mostfreq	multi	coll	0.37	0.14	**0.79**	0.31	*0.35*	0.48	0.19	0.28

Table 4: APPENDIX — baseline results for target term "boc" (maximum and minimum values highlighted in bold and italics, respectively)

Training algorithm	**Classification type**	**Vector type**	**Accuracy**		**Precision**		**Recall**		**F1**	
			Avg	Std Dev	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
nb	A vs. not A	bow	0.73	0.13	0.74	0.25	0.77	0.21	0.71	0.16
nb	A vs. not A	coll	0.79	0.14	0.81	0.22	0.73	0.31	0.71	0.26
nb	B vs. not B	bow	0.67	**0.19**	0.69	0.36	0.74	0.30	0.60	0.27
nb	B vs. not B	coll	0.75	0.17	0.71	0.35	0.65	0.38	0.61	**0.36**
nb	D vs. not D	bow	**0.84**	0.10	**0.85**	*0.15*	**0.83**	*0.18*	**0.82**	*0.12*
nb	D vs. not D	coll	0.82	0.13	0.82	0.20	0.82	0.20	0.80	0.17
nb	multi	bow	0.63	0.16	0.65	0.37	0.62	0.35	0.56	0.33
nb	multi	coll	*0.60*	0.17	*0.64*	0.35	*0.58*	0.37	*0.52*	0.31
me	A vs. not A	bow	0.75	0.12	0.73	0.28	0.73	0.27	0.69	0.25
me	A vs. not A	coll	0.79	*0.09*	0.81	0.20	0.71	0.31	0.70	0.25
me	B vs. not B	bow	0.66	0.17	*0.64*	**0.38**	0.72	0.30	0.58	0.29
me	B vs. not B	coll	0.74	0.14	0.76	0.27	0.63	**0.40**	0.58	0.34
me	D vs. not D	bow	0.81	0.14	0.81	0.23	0.82	0.21	0.78	0.20
me	D vs. not D	coll	0.76	0.15	0.79	0.22	0.77	0.23	0.75	0.18
me	multi	bow	0.65	0.18	*0.64*	0.32	0.67	0.29	0.61	0.28
me	multi	coll	0.71	0.14	0.75	0.29	0.62	0.39	0.57	0.34

Table 5: APPENDIX — detailed results for target term "boc" (maximum and minimum values highlighted in bold and italics, respectively)