university of groningen / faculty of arts

# Source language prediction

## A part of speech based approach

Ian Matroos

**Bachelor thesis**
Informatiekunde
Ian Matroos
s2996588
June 21, 2018

# ABSTRACT

In this thesis, I describe a system used to predict the source language from a possibly translated English text. The texts that I use were originally produced in Dutch, English, French, German, Italian, or Spanish, and have been translated to English by native English speakers.

The most informative features that this system uses may be used to improve human and machine translation. This can be done by substituting some of the phrases containing part-of-speech n-grams that are indicators of the source language by alternative phrases that use n-grams that are more common in the target language. My approach is unique because it focuses on using genre independent features.

After training my system on a balanced dataset (containing 715 texts per language, for a total of 4290 texts), I get a ten-fold cross-validated F1-score of 0.69 on a selected part (see section 3.1) of the Europarl corpus (Koehn, 2005; Tiedemann, 2012). This score is achieved by using bigrams to 5-grams of part-of-speech tags as features. Using token n-grams (unigrams and bigrams), I get a ten-fold cross-validated F1-score of 0.88.

By training on an unbalanced dataset (containing 20910 texts in total), the F1-scores are higher for both the part-of-speech based approach and the token-based approach (0.74 and 0.93 respectively). However, the unbalanced systems score badly on the Books dataset (Tiedemann, 2012). On this dataset, the unbalanced part-of-speech based system gets an F1-score of 0.34, while the unbalanced token based system gets an F1-score of 0.33.

When evaluating my balanced systems on the Books dataset, I get a surprisingly high F1-score of 0.74 for the part-of-speech based system. The balanced token-based system also performed better than I expected, and got a F1-score of 0.69. I believe that the score of the part-of-speech based system may be higher than the training set because every document contains a larger amount of text, and because there are relatively few (39) relevant documents in the Books dataset.

The Books dataset contained no Dutch books, 19 English books, 16 French books, 2 German books, and one Italian and Spanish book. This means that the Books dataset is completely unbalanced, and that not every source language is present in this testset. This might have influenced the out-of-genre results.

# CONTENTS

# PREFACE

As I am writing this thesis, I am wondering where the last three years went. When I started as a first year student, I liked programming, disliked almost all language courses, and I hoped that I didn't have to go through another math class for a while. As Information Science is quite a practical study, and less theoretical and math based than other options, I decided that I was just going to put up with the linguistic courses, and enjoy the other ones.

After somehow getting through 'Taalkunde voor CIW en IK', I was happily surprised that I had not only gotten through it on my first try, but that I actually seemed to understand the material. It wasn't long before I decided to follow the course 'General Linguistics', as it was a recommended course for students who picked the Artificial Intelligence minor. I wasn't convinced that it was going to be a fun course, but at least I felt like I should be able to get through it with the knowledge that we gained from Taalkunde. Again, I was happily surprised. The lecturer that we had for the course was one of the most enthusiastic lecturers I have had during these past three years. She not only explained the material in a way that was easy to follow, but she actually got some students enthusiastic about linguistics as well.

After my first year of Information Science, I was certain that with my love for programming, and newfound enthusiasm for linguistics, I was going to enjoy the next years for sure. I became a teaching assistant for a bunch of different courses during my second and third year, and have been given the privilege of becoming one of the co-authors of a scientific paper.

I would like to thank the following people who have made this possible:

- Mike Huiskes and Jennifer Spenader, who made me enjoy linguistics
- All professors and lecturers who have given me the opportunity to be a teaching assistant
- Rob van der Goot, Nikola Ljubešić, Malvina Nissim, and Barbara Plank, who have given me the privilege of being a co-author of their paper
- Antonio Toral, who has supervised this thesis

I also want to thank my parents, who have supported me throughout my entire education, and my brother.

# 1 INTRODUCTION

In this thesis, I will dive into the field of source language prediction. Since it has been found that there are significant differences between translated text and non-translated text (Frawley, 1984; Bernardini and Baroni, 2006), a lot of work has gone into finding these differences. There are also a few researchers who found that there wasn't just a difference between translated and non-translated text, but that this difference is for some part dependent on the languages involved. For example, Toury (2012) states that "in translation, phenomena pertaining to the make-up of the source text tend to force themselves on the translators and be transferred to the target text".

I want to find out if it is possible to predict the source language of a text based on part-of-speech tags, by building a system that can classify the source language of a text that has either been translated to English, or was already in English. The source languages I will be working with will be Dutch, English, French, German, Italian, and Spanish. These languages were also used by Halteren (2008), and are some of the most common source languages in Europarl. Of the 86,448 texts with an annotated language, 56,130 texts are annotated with one of these six languages (approximately 65%).

Since part-of-speech taggers are already available for more than 70 languages (currently 73, counting only those available in Universal Dependencies[1]), a system like this should be able to work on a lot of different target languages, without a lot of modifications. I hypothesize that my part-of-speech based approach will be more general than a token-based approach, which I will test by running the system on an out-of-genre testset. I will also look at what the most informative features are for each language, since these might be used to make human translated (and maybe even machine translated) text more natural.

If I manage to create a system that can predict the source language of a possibly translated text somewhat reliably, then this will indicate that there are different dialects of translationese. This means that a translated text will not only differ from a naturally written text, but that there will also be differences based on the language that the text was originally written in.

While a system like this might not be extremely practical, it will give us more insight in the differences in translated texts. A possible practical application might be the detection of plagiarism, specifically in the cases where a student has simply translated a paper or article into their native language. These cases can currently be hard to detect, as there is almost no lexical similarity between the translated text and the original. If my system is reliable enough, it might be used to indicate that a paper should be (automatically) translated into a certain language, and that that translation should be put trough the plagiarism checker as well. Without a system like mine, this would be impractical, as you would have to translate every paper into every commonly spoken language to do the same check.

I will first go over some relevant background literature in the next part of this thesis. Then I will discuss how I collected and processed my data; what my classification pipeline looks like and what features it uses; how well my systems perform in- and out-of-genre; and finally I will answer my research question, and give some suggestions for future work.

[1] http://universaldependencies.org/

# 2 PREVIOUS WORK

Most literature about the topic of source language prediction seems to be not directly about the topic itself, but about native language identification. Since the data I will use contains only translations by native speakers of the target language, some features used by others might not work for this specific task.

The research of Halteren (2008) seems to be closely related to the topic of source language prediction. He created a system that can predict the native language of a text based on either a translation, the original text, or the text in all possible languages (Dutch, English, French, German, Italian, and Spanish). While he did not give a confusion matrix for every language he has worked with, he reported the accuracy per class. For English, he gets accuracies between 60.4% (using marker based classification) and 86.8% (using support vector regression). Halteren (2008) uses token-based features, which will most likely result in higher scores on the in-genre dataset than my part-of-speech tag based approach. In contrast, I expect my approach to work better on an out-of-genre testset.

Halteren (2008) indicates that there are dialects of translationese which differ based on the source language. This was also found by Lembersky et al. (2012), who tested this hypothesis. Lembersky et al. (2012) have found that language markers based on texts translated into a language similar to the original one work better than language markers based on texts translated into a less similar language. This could mean that similar languages will also produce similar dialects of translationese. If this is the case, my results might show lower scores between similar languages than between dissimilar languages.

Wong and Dras (2011) have used partial parse rules as a feature for native language identification. In their paper, they note that some of the important rules are grammatical in nature, and show constructions that are common in the native language of a writer. Since the texts in my dataset are translated by native speakers of the target language, these types of characteristics will most likely not be relevant for my research. In principle, translators will be translating into their native language, which means that all of these features should be representative of English in my case.

Looking at the features that are popular in native language identification tasks (Tetreault et al., 2013), we can see a few other features which might prove useful in (cross-genre) source language prediction. For cross-genre classification the Tree Substitution Grammars and Context Free Grammars seem interesting. For in-genre prediction, a combination of n-grams of tokens and characters might be interesting to try.

# 3 DATA AND MATERIAL

## 3.1 COLLECTION

Since I needed a large amount of human-translated text, and English untranslated text, I decided to use parts of the Europarl corpus (Koehn, 2005; Tiedemann, 2012). For each of the languages I am interested in (Dutch, English, French, German, Italian, and Spanish), I have first downloaded all speeches. Ultimately however, I only used the speeches with a length between 380 and 2500 tokens ($\mu = 622.79, \sigma = 290.59$, median $= 524$), to make my results as comparable as possible with Halteren (2008). To count the number of tokens, I have used NLTK (Loper and Bird, 2002).

## 3.2 ANNOTATION

The XML files in the Europarl corpus contain an attribute for speakers which contains the language that they were speaking. This attribute however, is often missing and sometimes incorrect. This is why I have removed all data that had either missing or incorrect information. Another problem with the data is that some speakers decide to address the parliament in English, instead of their native language. This might also occasionally happen for other languages, but after checking the data, I believe that this is extremely uncommon.

Since the dataset did not have any information about the country that a speaker was representing, or any other data that might help separate native speakers from non-native speakers, I have filtered the English speeches based on the name of the speaker. To do so, I have created a list of members of the European parliament (MEPs) for the United Kingdom between 1999 and 2014, which can be found in Appendix A. I then performed a check to see if the speaker's name was similar to one of the names in this list (using the whoswho python library[1]). This does mean that I am missing speeches from Irish MEPs, and any other group of MEPs that has English as a native language.

The names in the Europarl corpus are really inconsistent (see Figure 1). Sometimes the name denotes the last name of a speaker (e.g. "Speroni"), sometimes it denotes a function (e.g. "President"), a title (e.g. "Graefe zu Baringdorf"), and sometimes it is the full name of a speaker, either in firstname-lastname, lastname-firstname, or basically any other format (e.g. "Simpson Brian"). Sometimes, the political party that a speaker represents is also put in the name field, after the name. This is why I have performed a check if names are similar, instead of checking if they are the same.

Out of the 3302 English texts that are produced by a MEP from the United Kingdom according to this check, 7 are definitely not produced by a MEP from the United Kingdom. There are an additional 24 texts where only the first or last name was given, without any other identifying information (e.g. Donnelly, Martin, Kinnock). This means that of the texts that I use for English, less than 1% is not guaranteed to be produced by a native speaker. I assume that a lot of English texts have been missed by this check, but believe that it it is better to have a smaller amount of good data, than a larger amount of mediocre data.

[1] https://github.com/rliebz/whoswho

```
<SPEAKER ID="1" NAME="President">
<SPEAKER ID="2" LANGUAGE="IT" NAME="Speroni">
<SPEAKER ID="3" NAME="President">
<SPEAKER ID="4" NAME="President.">
<SPEAKER ID="5" NAME="McKenna">
<SPEAKER ID="6" NAME="Doyle">
<SPEAKER ID="7" NAME="President">
<SPEAKER ID="8" NAME="Simpson Brian">
<SPEAKER ID="9" NAME="President">
<SPEAKER ID="10" NAME="Cashman">
<SPEAKER ID="11" NAME="De Rossa">
<SPEAKER ID="12" NAME="Doyle">
<SPEAKER ID="13" NAME="President">
<SPEAKER ID="14" LANGUAGE="ES" NAME="Redondo Jiménez">
<SPEAKER ID="15" NAME="McCartin">
<SPEAKER ID="16" LANGUAGE="FI" NAME="Pesälä">
<SPEAKER ID="17" LANGUAGE="FR" NAME="Souchet">
<SPEAKER ID="18" LANGUAGE="EL" NAME="Baltas">
<SPEAKER ID="19" LANGUAGE="DA" NAME="Busk">
<SPEAKER ID="20" LANGUAGE="DE" NAME="Graefe zu Baringdorf">
<SPEAKER ID="21" NAME="Wallström">
<SPEAKER ID="22" LANGUAGE="ES" NAME="Redondo Jiménez">
<SPEAKER ID="23" NAME="Wallström">
<SPEAKER ID="24" NAME="President">
<SPEAKER ID="25" LANGUAGE="IT" NAME="Fatuzzo">
<SPEAKER ID="26" NAME="President">
<SPEAKER ID="27" LANGUAGE="DE" NAME="Mayer, Xaver">
<SPEAKER ID="28" NAME="President">
<SPEAKER ID="29" LANGUAGE="DE" NAME="Mayer, Xaver">
<SPEAKER ID="30" NAME="President">
```

**Figure 1:** Examples of speaker tags in the Europarl corpus

## 3.3 PROCESSING

I have used NLTK (Loper and Bird, 2002) to tokenize individual texts and to convert the tokens to part-of-speech tags. You can see two lines of a preprocessed file in Figure 2.

To convert a file to part-of-speech tags, I loop through every line of each file, tokenizing and tagging the files line-by-line. This was done because some annotation is present in some files (e.g. "(applause)"). This annotation is often on its own line, and I wanted to make sure that the tagger was not influenced by annotation sentences.

| Original text | Part-of-speech version |
|---|---|
| Mr President , first of all , I should like to thank you on behalf of the Group of the Alliance of Liberals and Democrats for Europe for your inaugural speech . | NNP NNP , RB IN DT , PRP MD VB TO VB PRP IN NN IN DT NNP IN DT NNP IN NNP CC NNPS IN NNP IN PRP$ JJ NN . |
| Our group can fully identify with the programme you announced . | PRP$ NN MD RB VB IN DT NN PRP VBD . |

**Figure 2:** Preprocessing example

# 4 METHOD

## 4.1 CLASSIFICATION PIPELINE

In order to train a system to predict the source language of a text, I have created a classification pipeline that converts text files to vectors using the scikit-learn python library (Pedregosa et al., 2011), and then trains a classifier based on those vectors (using the same library). This pipeline consists of three parts: a vectorizer, a transformer and the classifier itself. I will now go into the details of each of these parts. Please note that all of my code can be found on GitHub[1].

### 4.1.1 The vectorizer

In order to convert my files into vectors, I have used scikit-learn's CountVectorizer. This vectorizer will normally create a sparse matrix that contains the number of times that a token occurs in a text. I have changed the tokenization part of this vectorizer to create strings representing n-grams instead of tokens. I have also set the vectorizer to create a binary matrix instead of using token counts. This was done because the length of the texts is highly variable, which would result in vastly different counts for the features within a class.

### 4.1.2 The transformer

Since the n-grams that do not seem to occur in a lot of texts are often good indicators of a certain class, I decided to use a tf-idf transformer. Since the data that we put in is actually binary data, this transformer will give us the inverse document frequency instead of the tf-idf score.

### 4.1.3 The classifier

Using the scikit-learn "Choosing the right estimator" map[2], I decided to use a linear support vector classifier. This classifier is implemented in scikit-learn in such a way that it can handle multiple classes using a one-vs-rest scheme. Since Halteren (2008) had good results using a SVC using the 'rbf' kernel, I also wanted to try it. I found however, that it did not scale well to my number of features at all. This made grid search almost impossibly slow and since Halteren (2008) did not report his C and $\gamma$ values, I found that an SVC with 'rbf' kernel was not something that I could use properly in the time frame of this thesis.

## 4.2 EVALUATION

To test my classification pipeline, I have used ten-fold cross-validation (keeping the ratios of classes the same in each fold). In order to make sure that the inverse document frequencies were not using information from the documents in the testset, I have implemented my system in such a way that it has to calculate the idf vectors separately for every fold.

[1] https://github.com/imatr/Source-language-prediction
[2] http://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

### 4.2.1 System output

For every class, my system calculates and reports the precision, recall, and F1-score. It does so for every fold, and for the combined output over all folds. It will also show a confusion matrix and the most informative features per class.

### 4.2.2 Balancing

Since my dataset is not balanced, the system might be biased toward certain classes. Because I want to validate my dataset against an out-of-genre testset, I have decided to train my system twice. Once with all available data, which I will balance by adjusting the weights in the classifier, and once with balanced data. I expect that the balanced system will work better on an out-of-genre testset, since it will not give higher weights to features of classes that occur less often in the testset, which might cause overfitting.

### 4.2.3 Out-of-genre testset

To evaluate my systems performance, I will run it on the English texts in the books corpus (Tiedemann, 2012), which are either untranslated or translated from one of my languages of interest into English. Of the 42 books that are available in English in this corpus, 39 fall into this category. A list of these books and their source language can be found in Appendix B. I will also train and evaluate a system on token n-grams, in order to see if my approach is indeed more general than a token based approach.

## 4.3 FEATURES

As described in the previous section, I have used n-grams of part-of-speech tags as features. In order to determine which n-grams were the best features, I have ran my system multiple times, using a different combination of n-grams each time (see Table 1 and Table 2). As we can see here, the highest scores are achieved when using either 5- or 6-grams and some combination of lower n-grams, but overall 5-grams seem to work slightly better than 6-grams. We can also see that 4-grams seem to work best when only using unigrams. For both the unbalanced and balanced dataset we can see that unigrams to 5-grams and bigrams to 5-grams seem to work best. I have decided to train my final systems using bigrams to 5-grams, because the lower amount of features will lower the memory footprint of the system slightly, and because the extra features don't improve the system.

### 4.3.1 Explored features

Wong and Dras (2011) found that parse rules were a good feature for native language identification. As source language prediction is a similar topic, I have tried to see if parse rules are an interesting feature for me as well. Instead of parsing the sentences into phrases, I have parsed them into universal dependencies using the SpaCy python library (Honnibal and Montani, 2017). I decided to use universal dependencies since I was more familiar with the implementation. Also, Kunilovskaya and Kutuzov (2017) have used universal dependency probabilities as a feature for predicting translator competency, and it seemed to work quite well for that task.

I have tried both universal dependency rules, and universal dependency rules based on part-of-speech tags, and both of these did not seem to work for me. I have only tested them on my old dataset with all files included for English (since I hadn't noticed that not all speakers where Native speakers when I tried this feature), and on that set (which contained 2155 files per

| Feature | F1-score | Features | F1-score | Features | F1-score | Features | F1-score |
|---|---|---|---|---|---|---|---|
| 1-grams | 0.29 | | | | | | |
| 2-grams | 0.57 | 1-2-grams | 0.57 | | | | |
| 3-grams | 0.68 | 1-3-grams | 0.69 | 2-3-grams | 0.69 | | |
| 4-grams | 0.72 | 1-4-grams | 0.73 | 2-4-grams | 0.73 | 3-4-grams | 0.73 |
| 5-grams | 0.70 | 1-5-grams | 0.74 | 2-5-grams | 0.74 | 3-5-grams | 0.74 |
| 6-grams | 0.63 | 1-6-grams | 0.74 | 2-6-grams | 0.74 | 3-6-grams | 0.73 |
| 7-grams | 0.53 | 1-7-grams | 0.73 | 2-7-grams | 0.73 | 3-7-grams | 0.72 |

**Table 1:** F1-scores over 10 folds using all data

| Feature | F1-score | Features | F1-score | Features | F1-score | Features | F1-score |
|---|---|---|---|---|---|---|---|
| 1-grams | 0.24 | | | | | | |
| 2-grams | 0.53 | 1-2-grams | 0.53 | | | | |
| 3-grams | 0.63 | 1-3-grams | 0.63 | 2-3-grams | 0.63 | | |
| 4-grams | 0.66 | 1-4-grams | 0.67 | 2-4-grams | 0.67 | 3-4-grams | 0.67 |
| 5-grams | 0.62 | 1-5-grams | 0.69 | 2-5-grams | 0.69 | 3-5-grams | 0.68 |
| 6-grams | 0.57 | 1-6-grams | 0.68 | 2-6-grams | 0.68 | 3-6-grams | 0.67 |
| 7-grams | 0.50 | 1-7-grams | 0.67 | 2-7-grams | 0.67 | 3-7-grams | 0.66 |

**Table 2:** F1-scores over ten folds after balancing the data

language, after balancing) I got an average F1-score of 0.43 for both approaches. This is why I decided not to pursue this feature.

### 4.3.2 Different part-of-speech tags

I have tried two different tagsets for my part-of-speech tags. The first is NLTK's default tagset, which consists of the Penn Treebank tagset[3] and some punctuation tags (45 tags in total). I have also tried NLTK's universal tagset, since I hoped that it would be more general, as it consists of only twelve different tags. This system using the universal tagset however, got an average F1-score of 0.59 (using bigrams to 5-grams). This could be caused by the fact that the tags are less specific, making unique combinations of tags less common.

### 4.3.3 Using tokens

As I hypothesized that my part-of-speech based approach will be more general than a token-based approach, I have also created two token-based systems. One uses the balanced dataset, while the other uses the unbalanced dataset. I have found that using only unigrams and bigrams resulted in the highest F1-score for my token-based systems. When cross-validating, the balanced system gets an F1-score of 0.88 (precision: 0.88, recall: 0.88) and the unbalanced system gets an F1-score of 0.93 (precision: 0.93, recall: 0.93).

That these scores are higher than the part-of-speech based scores is easily explained by looking at the most informative features in Figure 3. There we can see that the most informative features often contain the name of the country where a language is spoken, or a unigram denoting which language is being spoken. These language spoken markers do not consistently occur, and are sometimes used to show that a speaker only speaks a few sentences in that language. Since I only use the token based systems to see if my approach is indeed more general than a token based approach, I have not filtered these markers out.

[3] http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

| DE | EN | ES |
|---|---|---|
| (’president,’, ’ladies’) | (’the’, ’eu’) | (’i’, ’believe’) |
| (’let’, ’me’) | (’across’,) | (’community’,) |
| (’here.’,) | (’eu’,) | (’amongst’,) |
| (’and’, ’gentlemen,’) | (’behalf’,) | (’the’, ’spanish’) |
| (’gentlemen,’,) | (’behalf’, ’of’) | (’going’, ’to’) |
| (’ladies’, ’and’) | (’on’, ’behalf’) | (’(es)’, ’mr’) |
| (’ladies’,) | (’-’, ’madam’) | (’furthermore,’,) |
| (’(de)’, ’mr’) | (’group.’,) | (’-’, ’(es)’) |
| (’-’, ’(de)’) | (’group.’, ’-’) | (’spanish’,) |
| (’(de)’,) | (’-’, ’mr’) | (’(es)’,) |

| FR | IT | NL |
|---|---|---|
| (’(fr)’, ’madam’) | (’feel’, ’that’) | (’the’, ’netherlands,’) |
| (’shall’,) | (’president,’, ’ladies’) | (’great’, ’deal’) |
| (’i’, ’shall’) | (’italy’,) | (’number’,) |
| (’enable’,) | (’i’, ’feel’) | (’after’, ’all,’) |
| (’france,’,) | (’italy,’,) | (’number’, ’of’) |
| (’several’,) | (’(it)’, ’mr’) | (’dutch’,) |
| (’french’,) | (’the’, ’italian’) | (’a’, ’number’) |
| (’(fr)’, ’mr’) | (’-’, ’(it)’) | (’this.’,) |
| (’-’, ’(fr)’) | (’italian’,) | (’-’, ’(nl)’) |
| (’(fr)’,) | (’(it)’,) | (’(nl)’,) |

**Figure 3:** The most informative features per language (tokens)

# 5 RESULTS AND DISCUSSION

## 5.1 SYSTEM METRICS

As noted in previous sections, I have trained four systems. Two of those use part-of-speech bigrams to 5-grams, and the other two use token unigrams and bigrams. Since my objective with this thesis was creating a classifier that predicts the source language of a text using features that will work across different genres, I will not go into the results of the token-based classifier in too much detail.

### 5.1.1 Confusion between similar languages

If we take a look at the confusion matrices (Table 3 and Table 5) for the part-of-speech based systems, one thing that we might notice is that the systems seem to confuse certain language pairs more often than others. This is especially visible for the balanced system. Some of the language pairs that are harder to classify are: German and Dutch, Spanish and French, French and Italian, and Spanish and Italian. This could possibly be explained by the fact that these languages belong to the same language families. Dutch and German are both West Germanic languages. French, Italian, and Spanish are all in the Romance family of languages. The fact that these languages are confused more often could be an indicator that the systems have successfully learned about the structure of these languages.

We might also notice that there is almost no confusion between English and Dutch or between English and German. This seems to contradict the theory that the system looks at similarities between languages, since English is also a West Germanic language. However, if we look at the classification metrics (Table 4 and Table 6), this can be explained. It seems that English is easier for the system to correctly classify (possibly because it is the only untranslated language). It could be possible that most of the cases where language families normally decide the class are handled by features that are indicative of translated/untranslated text for English.

| | DE | EN | ES | FR | IT | NL |
|---|---|---|---|---|---|---|
| DE | 5282 | 2 | 106 | 329 | 71 | 401 |
| EN | 57 | 567 | 38 | 19 | 6 | 28 |
| ES | 204 | 2 | 2469 | 504 | 168 | 110 |
| FR | 455 | 5 | 409 | 4201 | 211 | 176 |
| IT | 236 | 1 | 250 | 446 | 1220 | 40 |
| NL | 753 | 1 | 77 | 185 | 31 | 1850 |

**Table 3:** Confusion matrix for the unbalanced 2- to 5-gram system

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| DE | 0.76 | 0.85 | 0.80 | 6191 |
| EN | 0.98 | 0.79 | 0.88 | 715 |
| ES | 0.74 | 0.71 | 0.73 | 3457 |
| FR | 0.74 | 0.77 | 0.75 | 5457 |
| IT | 0.71 | 0.56 | 0.63 | 2193 |
| NL | 0.71 | 0.64 | 0.67 | 2897 |
| avg / total | 0.75 | 0.75 | 0.74 | 20910 |

**Table 4:** Classification report of the unbalanced 2- to 5-gram system

| | DE | EN | ES | FR | IT | NL |
|---|---|---|---|---|---|---|
| DE | 422 | 27 | 36 | 39 | 42 | 149 |
| EN | 6 | 679 | 11 | 4 | 4 | 11 |
| ES | 32 | 19 | 447 | 72 | 92 | 53 |
| FR | 46 | 19 | 100 | 363 | 113 | 74 |
| IT | 42 | 12 | 65 | 64 | 510 | 22 |
| NL | 96 | 17 | 19 | 25 | 20 | 538 |

**Table 5:** Confusion matrix for the balanced 2- to 5-gram system

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| DE | 0.66 | 0.59 | 0.62 | 715 |
| EN | 0.88 | 0.95 | 0.91 | 715 |
| ES | 0.66 | 0.63 | 0.64 | 715 |
| FR | 0.64 | 0.51 | 0.57 | 715 |
| IT | 0.65 | 0.71 | 0.68 | 715 |
| NL | 0.64 | 0.75 | 0.69 | 715 |
| avg / total | 0.69 | 0.69 | 0.69 | 4290 |

**Table 6:** Classification report of the balanced 2- to 5-gram system

### 5.1.2 English

If we compare the classification results for the unbalanced system (Table 4) and the balanced system (Table 6), we can see that English has the best F1-score for both systems. This could support our explanation about the system also learning to recognize if a text has been translated or not.

It is also interesting to see that the unbalanced system has an extremely high precision for English, while having a relatively low recall compared to the balanced system. This might be caused by overfitting, since there are relatively few English texts in the unbalanced dataset. This would cause the system to follow a very narrow definition for English texts, which would explain the extremely high precision.

### 5.1.3 Statistical significance

As we can see in Figure 4, all systems perform better than the baseline (choosing randomly). To see if the systems perform significantly better I have performed multiple one-sample t-tests comparing both balanced systems and the unbalanced part-of-speech based system to the appropriate baseline. For the unbalanced token-based system, I have performed a Wilcoxon signed rank test, as the scores where not normally distributed (see Figure 5). All systems were significantly better than the baseline ($\alpha = 0.05$). The results of these tests can be found in Table 7.

I decided to use choosing randomly as a baseline for both the balanced and the unbalanced approach. Usually, the baseline for an unbalanced system is the score that the system would get when only choosing the most common class. In this case however, this would result in a baseline of 0.14, which is lower than it would be when choosing randomly. This is caused by two things. Firstly, the F1-score punishes large differences between precision and recall. This makes the F1-score for the most common class only 0.46 when it is always chosen. Secondly, the most common class is not extremely common, as only around 30% of the unbalanced dataset falls into this class.

I have also performed a paired t-test comparing the 10-fold cross-validated F1-scores of the balanced systems. I found that the token-based system performs significantly better than the part-of-speech based system ($p = 1.42e-09$, Cohen's $d = 10.28$). To compare the unbalanced systems, I have performed a Wilcoxon signed rank test (as the F1-scores of the unbalanced token-based system are not normally distributed). This test also showed that the token-based

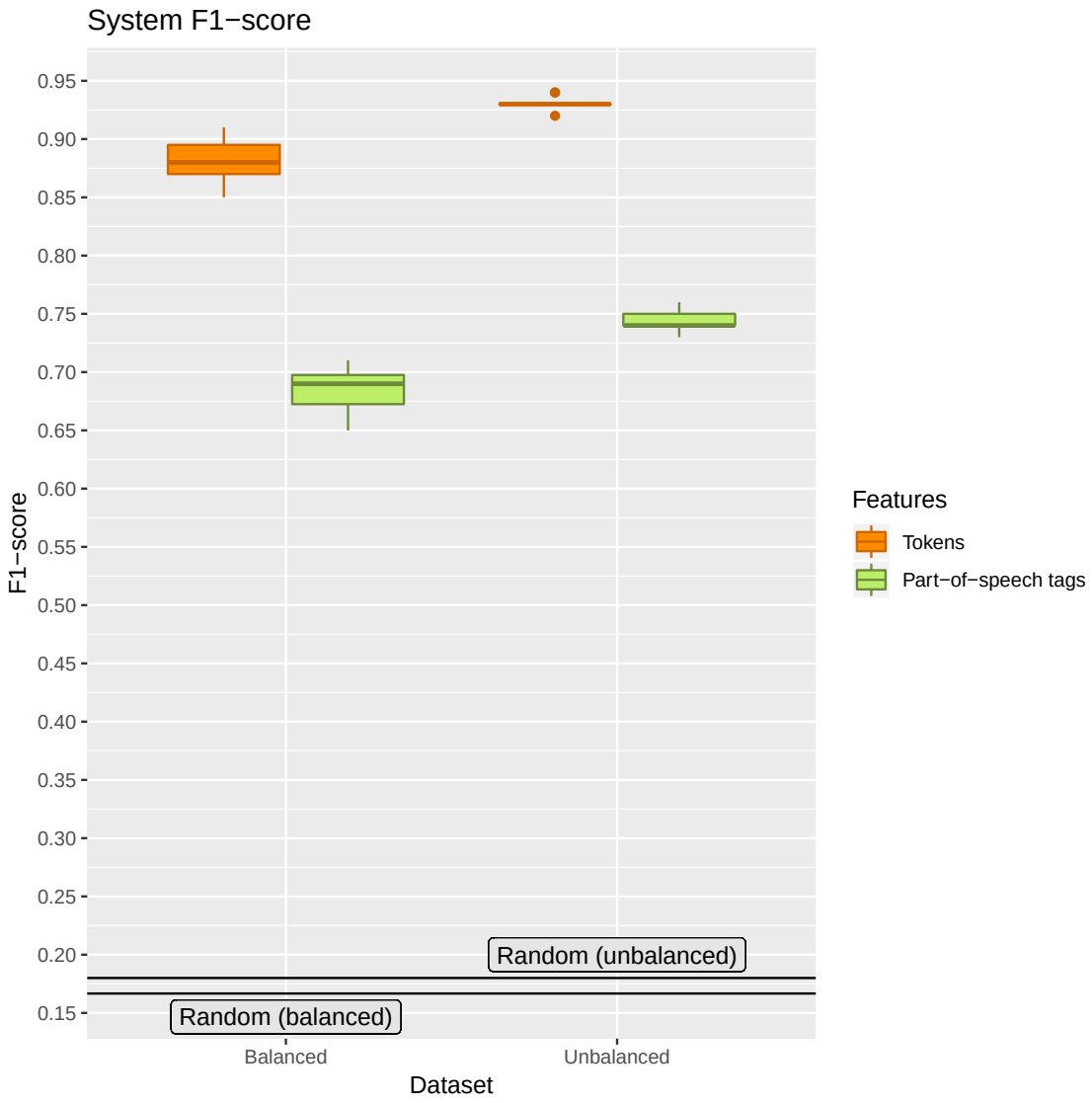


**Figure 4:** F1-scores of 10 fold cross-validation

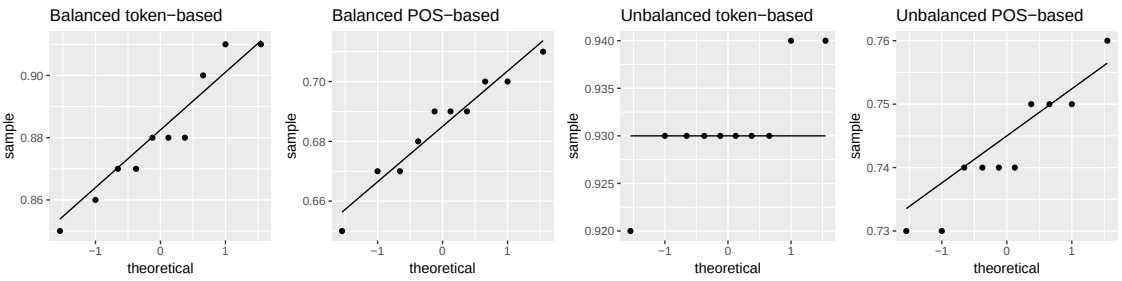


**Figure 5:** Q-Q plots showing the distribution of F1-scores when cross-validating

| System | p-value | Effect size |
|---|---|---|
| Balanced token-based | 9.48e-16 | Cohen's d: 35.28 |
| Balanced POS-based | 5.31e-15 | Cohen's d: 29.13 |
| Unbalanced token-based | 2.12e-3 | r: 0.97 |
| Unbalanced POS-based | < 2.2e-16 | Cohen's d: 59.35 |

**Table 7:** Statistical tests for cross-validation results

approach worked significantly better than the part-of-speech based approach ($p = 2.77e-03$). As I noted in subsection 4.3.3, this is to be expected.

## 5.2 IMPORTANT FEATURES

In Figure 6 we can see the ten most informative features per language (based on the balanced system). A larger list of 50 features per language can be found in Appendix C.

| DE | EN | ES |
|---|---|---|
| ('cc', 'nns', ',') | ('nnp', 'nnp', '.', ':') | ('in', 'nn', 'to', 'dt') |
| ('nns', 'cc', 'nns', ',') | ('.', ':') | ('.', 'nns', 'cc', 'nns', ',') |
| (',', 'nns', 'cc') | ('nnp', '.', ':', 'nnp', 'nnp') | ('in', 'nn', 'to', 'dt', 'nn') |
| (',', 'nns', 'cc', 'nns') | ('nnp', 'nnp', '.', ':', 'nnp') | (',', 'nnp', 'nnp', ',') |
| (',', 'nns', 'cc', 'nns', ',') | ('nnp', '.', ':', 'nnp') | ('prp', 'vbp', 'vbg', 'to', 'vb') |
| ('nnp', 'nnp', ',', 'nns') | (':', 'nnp', 'nnp') | ('vbp', 'vbg', 'to', 'vb') |
| ('nnp', ',', 'nns') | (':', 'nnp', 'nnp', ',') | ('cc', 'wdt') |
| ('nnp', 'nnp', ',', 'nns', 'cc') | ('.', ':', 'nnp', 'nnp', ',') | ('prp', 'vbp', 'vbg', 'to') |
| ('nnp', ',', 'nns', 'cc', 'nns') | ('.', ':', 'nnp', 'nnp') | ('vbp', 'vbg', 'to') |
| ('nnp', ',', 'nns', 'cc') | ('.', ':', 'nnp') | ('dt', 'in', 'prp') |

| FR | IT | NL |
|---|---|---|
| ('vbn', ',', 'in') | (',', 'jj', 'nn', 'in') | ('vbz', 'rb', 'jj', 'in', 'dt') |
| ('nns', 'in', 'dt', 'nn', 'in') | ('nn', ':', 'prp') | ('dt', 'nn', '.', 'dt') |
| ('vb', 'prp', '.') | ('nns', ':') | ('.', 'nn', 'to', 'vb') |
| ('nn', 'vbn', 'to') | (',', 'vbg', 'in') | ('.', 'dt', 'vbz', 'jj') |
| ('nn', 'in', 'prp$', 'nns') | (':', 'prp', 'vbp') | ('.', 'nn', 'to') |
| ('dt', 'nn', 'in', 'prp$', 'nns') | (':', 'dt') | ('.', 'nn', 'to', 'vb', ',') |
| ('in', 'prp$', 'nns', '.') | ('nnp', 'nnp', ',', 'nns') | ('nn', 'in', 'dt', '.') |
| ('prp$', 'nns', '.') | ('nnp', 'nnp', ',', 'nns', 'cc') | ('.', 'in', 'dt', ',') |
| ('dt', 'nn', 'in', 'nn', ',') | (')', 'nnp', 'nnp', ',', 'nns') | ('dt', '.') |
| (',', 'in', 'nnp', ',') | (':', 'prp') | ('in', 'dt', '.') |

**Figure 6:** The most informative features per language

### 5.2.1 Overlap

We can directly see that there is a lot of overlap between the features within a language. If we look at German for example, we can see that the most informative feature is ('cc', 'nns', ','). The next feature we see is ('nns', 'cc', 'nns', ','), which is the most informative feature following a noun (plural). We could either see this as unfortunate, or as fortunate. We can say that this is unfortunate as this means that some features are important enough that they can put whatever context they occur most in in the most informative features list as well. But we might also say that this is fortunate, since these features are apparently great indicators for the dialect of translationese that we get when translating from this source language.

### 5.2.2 Punctuation

Another notable result is that a lot of the most informative features contain at least some form of punctuation. It is well known that punctuation is a great feature for a lot of different tasks, ranging from author profiling (Rangel et al., 2015) to music genre prediction (Mayer et al., 2008). The fact that punctuation is also relevant in source language prediction could indicate that people will alter their writing style and structure when translating into their native language.

## 5.3 OUT-OF-GENRE RESULTS

### 5.3.1 Unbalanced part-of-speech based approach

The results of training the unbalanced part-of-speech based system on Europarl and evaluating it on the Books dataset can be found in Table 8. As we can see, the system has not found any books that it deemed to have English as a source language. This is most likely the result of overfitting, as there were fewer English texts than texts of other languages. Even though the weights have been adjusted when training the classifier, the classifiers definition to what defines an English text has become very narrow.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| DE | 0.13 | 1.00 | 0.24 | 2 |
| EN | 0.00 | 0.00 | 0.00 | 19 |
| ES | 0.00 | 0.00 | 0.00 | 1 |
| FR | 0.68 | 0.94 | 0.79 | 16 |
| IT | 0.00 | 0.00 | 0.00 | 1 |
| NL | 0.00 | 0.00 | 0.00 | 0 |
| avg / total | 0.29 | 0.44 | 0.34 | 39 |

**Table 8:** Results of evaluating the unbalanced system on the Books dataset

### 5.3.2 Balanced part-of-speech based approach

The results of training the balanced part-of-speech based system on all data in the balanced dataset can be found in Table 9. This system surprisingly performs better on the out-of-genre evaluation than when cross-validating. This could most likely be explained by the size and balance of the book corpus. Because the corpus is relatively small and unbalanced, the F1-score is based on a lower number of results, and therefore less precise. Still, we can clearly see that the balanced dataset performs better than the unbalanced approach.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| DE | 0.33 | 1.00 | 0.50 | 2 |
| EN | 0.85 | 0.58 | 0.69 | 19 |
| ES | 0.00 | 0.00 | 0.00 | 1 |
| FR | 0.83 | 0.94 | 0.88 | 16 |
| IT | 1.00 | 1.00 | 1.00 | 1 |
| NL | 0.00 | 0.00 | 0.00 | 0 |
| avg / total | 0.80 | 0.74 | 0.75 | 39 |

**Table 9:** Results of evaluating the balanced system on the Books dataset

### 5.3.3 Unbalanced token based approach

As we can see in Table 10, this classifier has completely missed the English class, just like the unbalanced part-of-speech based system. This is most likely the result of overfitting due to the small amount of data available for English.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| DE | 1.00 | 0.50 | 0.67 | 2 |
| EN | 0.00 | 0.00 | 0.00 | 19 |
| ES | 0.00 | 0.00 | 0.00 | 1 |
| FR | 0.55 | 1.00 | 0.71 | 16 |
| IT | 0.00 | 0.00 | 0.00 | 1 |
| NL | 0.00 | 0.00 | 0.00 | 0 |
| avg / total | 0.28 | 0.44 | 0.33 | 39 |

**Table 10:** Results of evaluating the unbalanced, token-based system on the Books dataset

### 5.3.4 Balanced token based approach

We can see in Table 11, this classifier doesn't suffer from the overfitting like the unbalanced classifier did. The average F1-score is only 0.19 lower than it was on the training set. This means that there is a decrease of approximately 27.5%.

Even though there is a decrease, it is interesting to see that a token-based approach seems to work unexpectedly well. My expectation was that this approach would not work at all, since the most informative features are either genre related (see Figure 3), or a unigram like '(de)', which denotes that a certain language was spoken. As these unigrams do not occur in other genres, these should not influence the scores too much when comparing the systems with the out-of-genre results.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| DE | 0.00 | 0.00 | 0.00 | 2 |
| EN | 0.92 | 0.58 | 0.71 | 19 |
| ES | 0.00 | 0.00 | 0.00 | 1 |
| FR | 0.71 | 0.94 | 0.81 | 16 |
| IT | 0.33 | 1.00 | 0.50 | 1 |
| NL | 0.00 | 0.00 | 0.00 | 0 |
| avg / total | 0.75 | 0.69 | 0.69 | 39 |

**Table 11:** Results of evaluating the balanced, token-based system on the Books dataset

### 5.3.5 Comparison

We can clearly see that out-of-genre classification with an unbalanced testset gets higher results when training with a balanced dataset than when training with an unbalanced dataset. This seems to be true for both the part-of-speech tag based approach and the token based approach.

We can also see that for the balanced datasets, the average F1-score is surprisingly high. The token based system can, even if its most informative features are genre specific, work out-of-genre as well. The part-of-speech tag based approach does even better, and seems to score better on the Books dataset than when cross-validating. This could possibly be explained by the following:

- The test set is small, which decreases the significance of the metrics. This means that some cases might have been classified correctly based on pure luck.

- A book contains more unique part-of-speech tags, which means that our classifier has more information to work with. It seems logical that text classification usually works better for longer texts than it does for shorter text.

While the F1-scores of the part-of-speech based system and the token based system are not too different in absolute units, there is quite a large difference when comparing the scores to the in-genre scores. The part-of-speech based system gains about 9% relative to to the cross-validation scores, while the token based system loses about 27.5%.

Unfortunately, these differences seem to be caused by the small size of the testset. When testing on ten non-overlapping subsets of the testset, the results varied greatly (as the number of samples is even smaller than before). I have performed a two sample t-test to see if the part-of-speech based system was significantly better than the token based system, and found that the difference was not significant ($p = 0.45$).

# 6 CONCLUSION

My original research question stated: is it possible to predict the source language of possibly translated English text by using part-of-speech n-grams. I also noted that I was going to check if this part-of-speech based approach would work better on an out-of-genre testset than a token based approach.

## 6.1 RESULTS AND LIMITATIONS

I think that we can safely say that creating a system that can predict the source language of a possibly translated English text based on part-of-speech tags is possible. Both of my part-of-speech based systems perform significantly better than the associated baselines, and the effect size is large for both systems as well.

We have also seen that a system trained on a balanced dataset works better for out-of-genre classification. The added training data that we can get from using an unbalanced dataset, and adjusting weights to compensate for this unbalancing causes overfitting. This makes these types of systems to perform badly when testing them on an out-of-genre testset. For the balanced systems, it seems like there is no loss of accuracy when using the part-of-speech based system, while the token-based approach does have this loss of accuracy. We must note however, that the out-of-genre testset is rather small (and completely unbalanced), and therefore less precise than a larger testset.

As the Europarl corpus currently stands, it is next to impossible to select only texts that were translated from a specific language into a specific target language. Language tags are often missing, and sometimes even conflicting. It is also currently not possible to easily check who said something, as speaker-ids do not seem to correctly line up across different languages. The name of the speaker is currently not given in a consistent format as well. Because of these reasons, there is a possibility that some of the files I used have been put into the wrong category when creating my dataset.

## 6.2 FUTURE WORK

As future work, a system like this could be tested against corpora of multiple genres to see if these results are reproducible in other genres as well. More research could also be done to see how the length of a file influences the accuracy of a system like this. Generally, larger texts seem to perform better, which means that evaluating on the Books corpus could have inflated the out-of-genre results.

More research could also be done to see why translators prefer to use certain words to translate from specific languages. Intuitively, one might say that this is caused by words that occur in the source language that can be translated to a similar word (or the same word) in the target language. For example, translations from French to English seem to keep the word 'madam', whereas other translations from the other languages do not seem to use that word as often. Interestingly, English texts do seem to contain the word 'madam' as well.

Creating a version of the Europarl corpus that has names consistent with the Europarl MEP directory, and corrected language tags would also be extremely useful for any future work in this field. A dataset like this might be useful for authorship attribution and author profiling tasks as well. Using Europarl-wide speaker-ids instead of file-wide ids would already be a big

improvement, as it would then be possible to attach a database containing information about every speaker. This information could contain information like the name, gender, age, country, political party, native language, and a few other things of every speaker (or every MEP).

Vanmassenhove and Hardmeier (2018) have already annotated several language pairs, sentence by sentence with the sentence's language, the date that the sentence was spoken, the euroID of the speaker, and the speakers name, date of birth, age, and gender. In their data however, the information about the speaker is duplicated for each sentence. I believe that storing all information about the speaker in a single place might be better, as it avoids data duplication. The age of the speaker could also be left out, as it can easily be calculated based on the date of birth of the speaker and the session date.

It would also be interesting to see if some of the methods mentioned by Tetreault et al. (2013) could be used for source language prediction. Especially the tree substitution grammars seem interesting to me, as they contain far more linguistic information than part-of-speech n-grams. The same could be said about the context free grammars as well. As my results already indicate that there is more confusion between languages in the same language family, it would be interesting to test if this confusion is caused by word ordering, or if translators are influenced in their grammatical choices by the source language. If these tests indicate that there are indeed grammatical differences between translations and original texts, even when translated by native speakers, these grammars could either be used independently, or in conjunction with n-grams.

# Appendices

# A | MEPS FOR THE UNITED KINGDOM (1999-2014)

Alan John Donnelly
Alyn Smith
Andrew Duff
Andrew Henry William Brons
Anthea McIntyre
Arlene McCarthy
Ashley Fox
Ashley Mote
Bairbre de Brún
Barbara O'Toole
Baroness Nicholson of Winterbourne
Baroness Sarah Ludford
Bashir Khanbhai
Bill Miller
Bill Newton Dunn
Brian Simpson
Caroline Jackson
Caroline Lucas
Catherine Bearder
Catherine Stihler
Charles Tannock
Chris Davies
Christopher Beazley
Christopher Heaton-Harris
Christopher Huhne
Claude Moraes
Dame Glenis Willmott
Daniel Hannan
David Campbell Bannerman
David Martin
David Robert Bowe
David Sumberg
Den Dover
Derek Roland Clark
Derek Vaughan
Diana Wallis
Diane Dodds
Edward McMillan-Scott
Elizabeth Lynne
Elspeth Attwooll
Eluned Morgan
Emma McClarkin
Eryl Margaret McNally
Eurig Wyn
Fiona Hall
Gary Titley
Geoffrey van Orden
George Lyon
Gerard Batten
Giles Chichester
Glenys Kinnock
Glyn Ford
Godfrey Bloom
Gordon J. Adam
Graham Booth
Ian Hudghton
Ian R.K. Paisley
Ian Twinn
Imelda Mary Read
Jacqueline Foster
James Elles
James L.C. Provan
James Nicholson
Jean Lambert
Jeffrey Titford
Jill Evans
Jim Allister
John Alexander Corrie
John Bowis
John Bufton
John Hume
John Purvis
John Stuart Agnew
John Whittaker
Jonathan Evans
Julie Girling
Kay Swinburne
Keith Taylor
Linda McAvan
Malcolm Harbour
Marina Yannakoudakis
Mark Francis Watts
Marta Andreasen
Martina Anderson
Martin Callanan
Mary Honeyball
Michael Cashman
Michael John Holmes
Mike Nattrass
Neena Gill
Neil Parish
Nicholas Clegg

Nick Griffin
Nicole Sinclaire
Nigel Farage
Nirj Deva
Pauline Green
Paul Nuttall
Peter Skinner
Phil Bennion
Philip Bradbourn
Philip Bushill-Matthews
Phillip Whitehead
Professor Sir Neil MacCormick
Rebecca Taylor
Richard A. Balfe
Richard Ashworth
Richard Corbett
Richard Howitt
Robert Evans
Robert Goodwill
Robert Kilroy-Silk
Robert Sturdy
Roger Helmer
Roger Knapman
Roy Perry
Sajjad Karim
Sharon Bowles
Simon Francis Murphy
Sir Graham Watson
Sir Robert Atkins
Stephen Hughes
Struan Stevenson
Syed Kamall
Terence Wynn
The Earl of Stockton
The Lord Inglewood
The Lord Nicholas Bethell
Theresa Villiers
Thomas Wise
Timothy Kirkhope
Trevor Colman
Vicky Ford
William (The Earl of) Dartmouth

# B BOOKS IN THE BOOKS CORPUS

| Title | Author | Source language |
|---|---|---|
| Prozess | Franz Kafka | German |
| Verwandlung | Franz Kafka | German |
| Pride and Prejudice | Jane Austen | English |
| Sense and Sensibility | Jane Austen | English |
| Jane Eyre | Charlotte Bronte | English |
| Alice in wonderland | Lewis Carroll | English |
| The spy | James Fenimore Cooper | English |
| Moll Flanders | Daniel Defoe | English |
| Robinson Crusoe | Daniel Defoe | English |
| Pickwick Papers | Charles Dickens | English |
| Adventures of Sherlock Holmes | Arthur Conan Doyle | English |
| A Study in Scarlet | Arthur Conan Doyle | English |
| Great Shadow | Arthur Conan Doyle | English |
| Hound of the Baskervilles | Arthur Conan Doyle | English |
| Rodney Stone | Arthur Conan Doyle | English |
| Sign of Four | Arthur Conan Doyle | English |
| Three Men in a Boat | Jerome K. Jerome | English |
| Naulahka | Rudyard Kipling | English |
| Fall of the House of Usher | Edgar Allan Poe | English |
| Ivanhoe | Walter Scott | English |
| Tom Sawyer | Mark Twain | English |
| Don Quijote | Miguel Cervantes | Spanish |
| Le grand Meaulnes | Fournier Alain | French |
| Dame aux Camelias | Alexandre Dumas | French |
| Trois Mousquetaires | Alexandre Dumas | French |
| Madame Bovary | Gustave Flaubert | French |
| Notre Dame de Paris | Victor Hugo | French |
| Pierre et Jean | Guy de Maupassant | French |
| Chartreuse de Parme | Stendhal | French |
| Rouge et le Noir | Stendhal | French |
| 20000 lieues sous les mers | Jules Verne | French |
| Forceurs de blocus | Jules Verne | French |
| Ile mysterieuse | Jules Verne | French |
| Tour du monde en 80 jours | Jules Verne | French |
| Voyage au Centre de la Terre | Jules Verne | French |
| Candide | Voltaire | French |
| Germinal | Emile Zola | French |
| Therese Raquin | Emile Zola | French |
| Principe | Niccolo Machiavelli | Italian |

| Title | Author | Source language |
|---|---|---|
| Quo Vadis | Henryk Sienkiewicz | Polish |
| Anna Karenina vol. 1 | Leo Tolstoy | Russian |
| Anna Karenina vol. 2 | Leo Tolstoy | Russian |

# C MOST INFORMATIVE FEATURES

## C.1 GERMAN

('nnp', 'nnp', ':', 'in')
('in', 'prp', '.', 'prp')
('nnp', 'vbz', 'rb', 'vbn', 'dt')
('in', 'prp', 'vbd', 'to', 'vb')
('dt', 'vbz', 'to', 'vb')
('nns', 'cc', 'rb')
('vb', 'prp', 'jj', 'in')
('rb', 'vb', 'vbn')
('dt', 'nn', 'in', 'in')
('prp', 'in', 'dt', 'nnp', 'nnp')
('rb', 'vbp', 'to')
('nns', 'md', 'rb', 'vb', 'vbn')
('in', 'prp', 'to')
('nn', 'nn', 'rb')
('rb', 'to', 'vb', 'vbn')
('nns', 'cc', 'nns')
(',', 'prp', 'vbp', 'rb', 'rb')
('jj', 'cc', 'nn')
('vbp', 'dt', 'rb')
('cc', 'rb', 'rb')
('rb', ',', 'cc')
('nnp', 'nnp', ',', 'nnp', ',')
('in', 'prp', 'vbp', 'to', 'vb')
('nn', 'vbz', 'to', 'vb', 'vbn')
('vb', 'vbn', 'jj')
('rb', 'vbp', 'to', 'vb')
('in', 'prp', 'vbd', 'to')
('jj', 'to', 'prp')
('.', 'vb', 'prp')
('vb', 'prp', 'jj')
('jj', 'nn', 'rb')
('md', 'rb', 'vb', 'vbn')
(',', 'nns')
('in', 'prp', '.')
('nnp', 'nnp', ',', 'nnp', 'nnp')
('nn', 'in', 'in')
('prp', 'jj', 'in')
('nnp', 'nnp', ',', 'nnp')
('vbz', 'to', 'vb', 'vbn')
('in', 'prp', 'in')
('cc', 'nns', ',')
('nns', 'cc', 'nns', ',')
(',', 'nns', 'cc')
(',', 'nns', 'cc', 'nns')
(',', 'nns', 'cc', 'nns', ',')
('nnp', 'nnp', ',', 'nns')
('nnp', ',', 'nns')
('nnp', 'nnp', ',', 'nns', 'cc')
('nnp', ',', 'nns', 'cc', 'nns')
('nnp', ',', 'nns', 'cc')

## C.2 ENGLISH

('nns', 'in', 'prp', 'vbp')
('in', 'nnp', 'nnps')
(':', 'nnp', 'nnp', ',', 'rb')
('nns', '.', 'prp', 'vbp', 'to')
('cc', 'prp', 'vbp', 'to', 'vb')
('prp', 'vbp', 'rb', 'to')
('dt', 'nns', 'in', 'prp')
('.', 'prp', 'vbp', 'dt')
('prp', 'vbp', 'dt', 'nn')
('nnp', 'nnp', 'nnp', '.', ':')
('vbp', 'rb', 'to')
('to', 'vb', 'jj', 'in')
('cc', 'prp', 'vbp', 'to')
('nnp', 'nnp', ',', 'prp')
('prp', 'vbp', 'dt', 'nnp')
('nnp', ':', 'nnp', 'nnp', '.')
(':', 'nnp', 'nnp', '.', ':')
(':', 'nnp', 'nnp', '.')
('nnp', 'nnp', 'nnp', 'nnp', '.')
('nn', 'in', 'dt', 'nnp', 'nnp')
('nns', 'vbp', 'vbd', ',')
('prp', 'vbp', 'to', 'vb', 'in')
('nns', 'vbp', 'vbd')
('nns', 'in', 'prp')
('.', 'prp', 'vbp', 'to', 'vb')
('vbp', 'to', 'vb', 'in')
('.', 'prp', 'vbp', 'to')
(':', 'nnp', 'nnp', ',', 'in')
('nnp', 'nnp', '.')
('dt', 'nnp', 'nnp', '.')
('in', 'dt', 'nnp', 'nnp', '.')
(':', 'nnp', 'nnp', ',', 'dt')
('in', 'nn', 'in', 'dt', 'nnp')
('nn', '.', ':')
('dt', 'nnp', 'nnp', '.', ':')
('nn', '.', ':', 'nnp')
('nn', '.', ':', 'nnp', 'nnp')
(':', 'nnp', 'nnp', ',', 'prp')
(':', 'nnp')
('nnp', '.', ':')
('nnp', 'nnp', '.', ':')
('.', ':')
('nnp', '.', ':', 'nnp', 'nnp')
('nnp', 'nnp', '.', ':', 'nnp')
('nnp', '.', ':', 'nnp')
(':', 'nnp', 'nnp')
(':', 'nnp', 'nnp', ',')
('.', ':', 'nnp', 'nnp', ',')
('.', ':', 'nnp', 'nnp')
('.', ':', 'nnp')

## C.3 SPANISH

('nnp', 'in', 'nn', 'to')
('wdt', ',')
('vbn', 'in', 'dt', 'nnp', 'in')
('in', 'dt', 'vbz', 'dt')
(',', 'prp', 'rb', 'vbp', 'in')
(',', 'in', 'nn', 'to', 'vb')
('wdt', 'prp', 'vbp')
('in', 'dt', 'nn', 'prp')
('nn', '.', 'nns', 'cc', 'nns')
('jj', 'nns', ',', 'in', 'dt')
('dt', 'nn', ',', 'nnp', 'nnp')
(',', 'fw')
('vbp', 'nns')
('vb', 'prp', 'to', 'vb')
('nnp', 'nnp', 'vbd', 'nn')
('dt', 'nn', 'prp', 'vbp', 'vbg')
('vbg', 'to', 'vb')
('prp', 'vbp', 'vbg')
('.', 'nns', 'cc')
(',', 'jj', 'in')
('.', 'in', 'nn', 'to', 'dt')
('in', 'nn', 'to')
('vbp', 'in', 'dt', 'vbz')
('jj', 'nn', 'in', 'nn', 'to')
('dt', 'in', 'dt', 'nns')
('nnp', 'nnp', ',', 'prp', 'vbp')

('nns', ',', 'jj', 'in')
('nn', 'in', 'nn', 'to')
('in', 'nn', 'to', 'vb')
('cc', ',', 'in', 'dt', 'nn')
(',', 'prp', 'md', 'rb', 'in')
('nnp', 'nnp', ',', 'prp')
('.', 'nns', 'cc', 'nns')
(',', 'jj', 'in', 'dt')
('vbz', 'to', 'vb', ',')
('in', 'dt', 'vbz')
('nnp', ',', 'prp', 'vbp')
('nn', 'prp', 'vbp', 'vbg')
(',', 'in', 'nn', 'to')
('vbz', 'vbg', 'to')
('in', 'nn', 'to', 'dt')
('.', 'nns', 'cc', 'nns', ',')
('in', 'nn', 'to', 'dt', 'nn')
(',', 'nnp', 'nnp', ',')
('prp', 'vbp', 'vbg', 'to', 'vb')
('vbp', 'vbg', 'to', 'vb')
('cc', 'wdt')
('prp', 'vbp', 'vbg', 'to')
('vbp', 'vbg', 'to')
('dt', 'in', 'prp')

## C.4 FRENCH

('md', 'in', 'nn')
('in', 'dt', 'nn', 'vbd')
('nn', 'in', 'nns', ',', 'in')
('dt', 'nn', 'wdt', 'vbz', 'rb')
('vb', ',', 'in')
('nn', ',', 'dt', 'nn', 'in')
('rb', 'vb', 'rb')
('dt', 'nn', 'in', 'prp$')
('in', 'dt', 'nn', 'in', 'vbg')
('jj', 'nnp', ',', 'nnp')
('dt', 'nn', 'vbn', 'in')
('nnp', ',', 'nnp', ',')
('.', 'wrb', 'md', 'prp')
('prp', 'jj', 'to')
('.', 'nnp', 'in')
('dt', 'nns', ',', 'nn')
('md', 'rb', 'vb', 'dt', 'jj')
('wrb', 'md', 'prp')
('.', 'wrb', 'md', 'prp', 'vb')
('nns', ',', 'wdt', 'vbp')
('dt', 'nn', 'vbn', 'in', 'dt')
('vb', ',', 'in', 'dt')
('.', 'prp', 'md', 'rb', 'vb')
('pos', '.')
('prp', 'vbp', ',')
(',', 'wp$')
('to', 'vb', 'prp', '.')
('vb', 'to', 'prp')
('rb', 'vb', 'prp', 'to')
('dt', 'nn', 'vbn')
('prp', 'vb')
('nn', 'in', 'jj', 'nn', '.')
('.', 'nn', ',')
('md', 'in')
('dt', 'nnp', '.', ':')
('in', 'dt', 'nnp', '.', ':')
('prp', 'jj', 'to', 'vb')
('dt', 'nnp', '.', ':', '(')
('nn', ',', 'dt', 'nns')
('vbg', 'to', 'vb')
('vbn', ',', 'in')
('nns', 'in', 'dt', 'nn', 'in')
('vb', 'prp', '.')
('nn', 'vbn', 'to')
('nn', 'in', 'prp$', 'nns')
('dt', 'nn', 'in', 'prp$', 'nns')
('in', 'prp$', 'nns', '.')
('prp$', 'nns', '.')
('dt', 'nn', 'in', 'nn', ',')
(',', 'in', 'nnp', ',')

## C.5 ITALIAN

('vbn', ':')
('rb', ',', 'vbg')
('dt', 'jj', 'nn', ':')
('nns', ',', 'vbg', 'in')
('nns', 'vbn', 'in')
(',', 'nns', 'cc')
('in', 'dt', 'jj', 'nns', ',')
('nn', ',', 'vbg', 'in')
('nns', ':', 'prp')
('nn', ':', 'prp', 'vbz')
('rb', 'jjs', 'in')
(',', 'prp', 'vbp', 'in')
('vbg', ':')
(':', 'in', 'dt')
('jj', 'cc', 'jj', ':')
(',', 'nns', 'cc', 'nns', ',')
(':', 'dt', 'nn')
(',', 'nns', 'cc', 'nns')
(',', 'jj', 'nn')
(':', 'vbg')
('nns', ',', 'vbg')
('vbn', 'in', 'nnp', 'cd')
(',', 'vbg', 'in', 'dt')
('jj', ',', 'jj', 'nn', 'in')
('nn', ':', 'dt')
(':', 'in')
('nnp', ',', 'nns')
('jj', ',', 'jj')
(':', 'rb')
(',', 'rb', 'jjs')
('nnp', ',', 'nns', 'cc', 'nns')
('jj', 'nn', ':')
('nnp', ',', 'nns', 'cc')
('dt', 'jj', ',', 'jj', 'nn')
(',', 'vbg')
('rb', 'jjs')
(':', 'prp', 'vbz')
(':', 'dt', 'jj')
('dt', 'jj', ',', 'jj')
('jj', ',', 'jj', 'nn')
(',', 'jj', 'nn', 'in')
('nn', ':', 'prp')
('nns', ':')
(',', 'vbg', 'in')
(':', 'prp', 'vbp')
(':', 'dt')
('nnp', 'nnp', ',', 'nns')
('nnp', 'nnp', ',', 'nns', 'cc')
(')', 'nnp', 'nnp', ',', 'nns')
(':', 'prp')

## C.6 DUTCH

('nn', 'in', 'dt', ',')
('rb', 'rb', '.')
('vbn', 'to', 'dt')
('dt', 'nn', '.', 'dt', 'vbz')
('nnp', 'nnp', '.', ':', '(')
('nn', 'in', 'wdt')
(',', 'in', 'nn', '.')
('.', 'md')
('nn', '.', 'jj')
('rb', 'jj', 'in', 'dt', 'nn')
('nnp', '.', 'prp', 'vbz')
('vb', 'to', 'vb', 'dt', 'jj')
('.', 'rb', 'rb', 'in')
('nn', 'rb', '.')
('in', 'dt', ',')
('nn', '.', 'in', 'dt', ',')
('in', 'prp$', 'nn', ',')
('dt', 'vbz', 'rb')
('jj', 'nn', 'in', 'dt', '.')
('.', 'rb', 'rb')
('in', 'jj', '.')
(',', 'cc', 'rb', 'in')
('nn', '.', 'in', 'prp$', 'nn')

('.', 'in', 'prp$')
('.', 'nnp')
('nn', '.', 'in', 'prp$')
('rb', 'jj', 'in', 'dt')
('dt', 'nn', '.', 'dt', 'nn')
('.', 'prp', 'md', 'vb', 'to')
('.', 'in', 'prp$', 'nn', ',')
('rb', '.')
('.', 'in', 'prp$', 'nn')
('.', 'dt', 'vbz', 'rb')
('dt', 'vbz', 'jj')
('rb', '.', 'dt')
('dt', 'md', 'rb', 'vb')
('vb', 'to', 'vb', 'dt')
('md', 'vb', 'to', 'vb', 'dt')
('dt', 'vbz', 'in')
('dt', 'md', 'rb')
('vbz', 'rb', 'jj', 'in', 'dt')
('dt', 'nn', '.', 'dt')
('.', 'nn', 'to', 'vb')
('.', 'dt', 'vbz', 'jj')
('.', 'nn', 'to')
('.', 'nn', 'to', 'vb', ',')
('nn', 'in', 'dt', '.')
('.', 'in', 'dt', ',')
('dt', '.')
('in', 'dt', '.')

# BIBLIOGRAPHY


Bernardini, S. and M. Baroni (2006). Spotting translationese: A corpus-driven approach using support vector machines. In *Corpus linguistics 2005*. Centre for Corpus Research.

Frawley, W. (1984). Prolegomenon to a theory of translation. *Translation: Literary, linguistic and philosophical perspectives 159*, 175.

Halteren, H. (2008). Source language markers in europarl translations. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pp. 937–944. Coling 2008 Organizing Committee.

Honnibal, M. and I. Montani (2017). spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT summit*, Volume 5, pp. 79–86.

Kunilovskaya, M. and A. Kutuzov (2017). Universal dependencies-based syntactic features in detecting human translation varieties. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories*, pp. 27–36.

Lembersky, G., N. Ordan, and S. Wintner (2012). Language models for machine translation: Original vs. translated texts. *Computational Linguistics 38*(4), 799–825.

Loper, E. and S. Bird (2002). Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, Stroudsburg, PA, USA, pp. 63–70. Association for Computational Linguistics.

Mayer, R., R. Neumayer, and A. Rauber (2008). Rhyme and style features for musical genre classification by song lyrics. In *Ismir*, pp. 337–342.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12*, 2825–2830.

Rangel, F., P. Rosso, M. Potthast, B. Stein, and W. Daelemans (2015). Overview of the 3rd author profiling task at pan 2015. In *CLEF*, pp. 2015. sn.

Tetreault, J., D. Blanchard, and A. Cahill (2013). A report on the first native language identification shared task. In *Proceedings of the eighth workshop on innovative use of NLP for building educational applications*, pp. 48–57.

Tiedemann, J. (2012, may). Parallel data, tools and interfaces in opus. In N. C. C. Chair), K. Choukri, T. Declerck, M. U. Dogan, B. Maegaard, J. Mariani, J. Odijk, and S. Piperidis (Eds.), *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Toury, G. (2012). *Descriptive Translation Studies and beyond: revised edition*, Volume 100. John Benjamins Publishing.

Vanmassenhove, E. and C. Hardmeier (2018). Europarl datasets with demographic speaker information.

Wong, S.-M. J. and M. Dras (2011). Exploiting parse structures for native language identification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, Stroudsburg, PA, USA, pp. 1600–1610. Association for Computational Linguistics.