Gender prediction using lexical, morphological, syntactic and character-based features in Dutch

Evgenii Glazunov

e.glznv@yandex.ru

National Research University Higher School of Economics, Moscow, Russia

Abstract. This work is a result of participation in shared task on gender detection in Dutch. The task was to predict gender within and across different genres. This work applies some existing ideas about using lexical and more abstract text representations (morphological, syntactical labels, text bleaching). It provides a comparison of different features across genres in two types of tasks and presents two pipelines. Using three types of features, we found that lexical features are more significant, although other features also show good results making the model more robust. Final scores where in range 0.61-0.64 for in-genre and 0.53-0.56 for cross-genre prediction.

1 Task and data

The task was to detect gender of the author in three genres: news, Twitter posts and Youtube comments, using training set of the same genre as test set (in-genre prediction) or different genre(s) (cross-genre prediction).

Training set size (genders are balanced within sample):

- 1832 news texts (≈ 340000 tokens)
- 20000 Twitter posts (≈ 380000 tokens)
- 14744 YouTube comments (≈ 280000 tokens)

2 Workflow

This work presents step-by-step description of creating a pipeline during work on shared task. Firstly, we discuss different text representations that may help to extract features of different nature. Then we describe feature extraction itself and some preliminary results on cross-validation using different representations and combined features. Finally, we present a comparison of two pipelines based on two feature combinations and comment final results on the test set.

Type of Text	Example
Original text	Deze video bekijken terwijl je een vrouw bent!
Lemmatized text	deze video bekijken terwijl je een vrouw bent!
POS	DET NOUN VERB SCONJ PRON DET NOUN NOUN PUNCT
Syntactic relations	det obj root mark nsubj det advcl nmod punct
CV-mask	cvcv cvcvv cvcvccvc cvccvcc cv vvc ccvvc cvcc!
UL-mask	ULLL LLLLL LLLLLLLL LLLLLLL LL LLL LLLLL LLLL!
Word length	4 5 8 7 2 3 5 4 !

Table 1. Different text representations

3 Preprocessing

To find what features characterize language of men and women we need to explore the different levels of the text from phonological or graphical to syntactic and semantic level. Preprocessing allows to create a text representation (a modified version of text that preserves its certain features and neutralize other ones). These representations correspond to different text levels and will be used later for feature extraction.

Some works (e.g. (van der Goot et al., 2018)) have shown that lexical features demonstrate the highest score. There is also a hypothesis that for cross-genre prediction we need to use more abstract features (not text specific, but independent e.g. character-based). Being so general, they may help us achieve higher results in cross-genre models because text-specific lexical features intuitively are more genre-specific. This needs to be checked in our experiments.

We use three groups of features in ascending order of their abstractness:

- Lexical
- Morphological and syntactic
- Character-based

Lexical features can be either tokens or lemmas. In our case, these are lemmas, since a relatively small amount of training set (both in token and documents) does not allow the use of tokens. The second group is presented by part-of-speech (POS) tags and labels of syntactic relations. The third group comprises text bleaching (van der Goot et al., 2018) features. These features are very abstract and in this work character-based. This makes them applicable for cross-genre prediction as well as for in-genre one. They are consonant/vowel mask (texts is a sequence of marks showing whether this character denotes vowel or consonant), upper/lower case mask and word lengths (in characters). Example of different text representations is demonstrated in Table 1.

Each language has its own characteristics, which are taken into account in specialized tools for working with it, but it is difficult for an external researcher (who has no experience with this language and does not speak this language) to find and use them properly. On the one hand, some lexical features may be selected and checked on credibility only working with semantics that requires

Representation	**News**	**Twitter**	**YouTube**
Lemmatized	0.665	0.622	0.598
POS	0.574	0.568	0.548
Syntax	0.579	0.553	0.538
CV-mask	0.625	0.587	0.606
UL-mask	0.598	0.590	0.583
Word length	0.578	0.58	0.584

Table 2. Accuracy score using text representations separately. Vectorizer: TfidfVectorizer. Classifier: LogisticRegression. Score: accuracy, 10-fold cross-validation

language knowledge. On the other hand, many NLP tasks (e.g. translation) can be solved without any knowledge of processed language(s). Therefore, in this paper we use tools available for a wide range of languages and that can be used without specific knowledge of Dutch: available pre-trained Word2Vec (W2V) (Mikolov et al., 2013) and UDPipe (Straka and Straková, 2017) models. The former is used for working on lexical features and the latter for lemmatization, part-of-speech tagging and extracting syntactical information.

4 Feature extraction

In the first experiment we want to compare usefulness of each type of representation. We use TF-IDF vectorizer from sklearn Python library as a feature extraction instrument and logistic regression as a classifier. This step is necessary because we need to determine the potential of each type of text representation and examine a range of accuracy scores that we can expect from our model on the final stage. We can use accuracy metric since the sample is balanced (50/50 texts by men and women). Table 2 shows scores gained using different text representations.

As we can see in Table 2, the most useful features are lemmatized text and abstract character-based masks. However, we cannot exclude other features because their scores are also good on this scale and they may show themselves better in later experiments.

The next step for working with vocabulary is to combine TF-IDF and Word2Vec features. This can be gained by multiplying matrices with TF-IDF and W2V vectors in order to get one vector for each text. The result vector of this multiplication corresponds to weighted (TF-IDF) sum of semantic vectors of words in each text.

5 The experiment with combined features

The next step is to combine all features in order to stabilize our model and make it more robust. As far as we might send two predictions for each task, we decided to try two combinations of features.

	News		**Twitter**		**Youtube**	
	P1	P2	P1	P2	P1	P2
In	0.651	0.642	0.619	0.607	0.617	0.616
Cross	0.537	0.534	0.561	0.55	0.533	0.536

Table 3. Accuracy score using text representations separately. Vectorizer: TfidfVectorizer. Classifier: LogisticRegression. Score: accuracy, 5-fold cross-validation for in-genre, heldout for cross-genre.

The first pipeline consists of following groups of features:

- TF-IDF (using words) with n-gram range from 1 to 4: W2V vectors with TF-IDF weights, POS tags, syntactic labels, CV-mask, UL-mask and word lengths
- TF-IDF (using characters) with n-gram range from 1 to 4: tokenized text, CV-mask, UL-mask
- TF-IDF + W2V vectors of lemmatized text

In case of POS tags and other non-lexical features TF-IDF vectorizer considered tags or labels as real words of metalanguage.

The second pipeline follows the first one, excluding lexical features (W2V). Table 3 presents a comparison of these two pipelines. We can see that the gap between the pipelines is very small, so using only abstract features gives a result comparable to complex features as W2V.

As we saw in Table 2, using only lexical features gives us a very good result, on news texts even better result. There are two reasons why we didn't go back to using exclusively this representation. The first one is the instability of vocabulary: we cannot guarantee that new data will follow the vocabulary we have, so the results are very unpredictable. The second one is homogeneity of our pipeline: our aim was to create a uniform pipeline for all genres, not separate models that show good results on cross validation.

6 Final results

Each task allowed two submissions (prediction from models of both pipelines). So we could test two combinations of features and check the hypothesis about robustness and quality of model based on more abstract features. For in-genre tasks, models were trained on sample of this particular genre while for cross-genre tasks we used two other genres, but not the target one (e.g. news + Twitter for Youtube comments).

	News		**Twitter**		**Youtube**	
	P1	P2	P1	P2	P1	P2
In	0.637	0.619	0.624	0.612	0.633	0.623
Cross	0.534	0.554	0.558	0.547	0.541	0.522

Table 4. Final accuracy score

The hypothesis that more abstract features (comparing with lexical ones) would be better in cross-genre tasks was refuted. We can see in Table 4 that the second pipeline achieved better results only in news genre. Probably, this happened because personal short text differs from the more formal one in terms of vocabulary. News texts represent a more literary language, while tweets and comments demonstrate more spoken language. Moreover, the themes of these texts are different.

7 Conclusion

In general, the results were lower than expected, although this may be because of the size of the sample or the weak gender differences in Dutch. As we can see in Table 4, a large gap exists between in-genre and cross-genre scores. Consequently, we can conclude that the features learned by models are rather genre-specific than general. We consider the model successful in terms of in-genre detection, because its score is above 0.6, while the score in case of cross-genre detection only slightly excels random choice.

8 Further work

This pipeline provides good results, but further work may include better feature selection. In our case there are thousands of numeric features obtained from different text representation, but we can expect that a lot of them are very noisy and have to be excluded from the result matrix of features. This can ameliorate existing model. Moreover, experiments with more sophisticated classifier and parameter selection may improve our score as well.

Acknowledgments

We would like to thank Malvina Nissim for a very inspiring talk about text bleaching on the conference in Nizhny Novgorod (Russia, Autumn 2018).

Bibliography

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.

Milan Straka and Jana Straková. 2017. Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics, Vancouver, Canada, pages 88–99.

Rob van der Goot, Nikola Ljubesic, Ian Matroos, Malvina Nissim, and Barbara Plank. 2018. Bleaching text: Abstract features for cross-lingual gender prediction. *CoRR* abs/1805.03122.