Exploring Classifier Combinations for Language Variety Identification

Tim Kreutz **Walter Daelemans**
CLiPS - Computational Linguistics Group
Department of Linguistics
University of Antwerp
{tim.kreutz,walter.daelemans}@uantwerpen.be

Abstract

This paper describes CLiPS's submissions for the Discriminating between Dutch and Flemish in Subtitles (DFS) shared task at VarDial 2018. We explore different ways to combine classifiers trained on different feature groups. Our best system uses two Linear SVM classifiers; one trained on lexical features (word n-grams) and one trained on syntactic features (PoS n-grams). The final prediction for a document to be in Flemish Dutch or Netherlandic Dutch is made by the classifier that outputs the highest probability for one of the two labels. This confidence vote approach outperforms a meta-classifier on the development data and on the test data.

1 Introduction

Discriminating between Dutch and Flemish in Subtitles (DFS) is a shared task at the VarDial evaluation campaign 2018. The task aims at identifying language variety in written Dutch texts, specifically subtitles from movies and television, and classifying them as either Netherlandic Dutch or Flemish Dutch (Zampieri et al., 2018).

Although DFS is organized for the first time at VarDial, it adheres to the workshop's general themes of closely related languages and language varieties. The long-running discriminating between similar languages (DSL) shared task has been organized previously, and has yielded lessons about distinguishing similar languages and language varieties (Zampieri et al., 2017).

Since the Netherlands and Flanders adhere to the same standard language (Dutch), the task at hand is one of language variety identification rather than similar language identification. This distinction is important because differences between language varieties are often less obvious than differences between different languages, however related they are (Goutte et al., 2016). Still, native Dutch speakers from the Netherlands or Belgium mostly have no problem coming up with anecdotal or typical differences, and linguistic work has previously formulated the most important distinctions.

There are plenty of practical applications for identifying Dutch language variety. It can for example extend current work on Author Profiling (AP) for Dutch by estimating a speaker's origin. In data selection, researchers might only be interested in texts written by either Dutch or Belgian authors and use an automated system to make the distinction. For the purposes of theoretical linguists, a machine learning system that outputs feature weights might indicate differences between the language varieties that have not been systematically studied before.

This paper will describe the submissions of the Computational Linguistics & Psycholinguistics (CLiPS) research center of the University of Antwerpen to the DFS shared task for VarDial 2018. As we alluded to in the introduction we will draw lessons from previous work that aimed at discriminating similar languages and language varieties, and more theoretical linguistic work on differences between Netherlandic Dutch and Flemish Dutch.

2 Related work

Eleven teams participated in the previous edition of the DSL shared task (Zampieri et al., 2017). The task involved determining the language or language variety for written news excerpts. It featured ten different languages with some having two or three varieties, making for a set of fourteen possible classes. For a multiclass classification problem with a random baseline at 10 percent, the performance of most systems was very good. The best system yielded a weighted F-score of .927 (Bestgen, 2017).

There are only slight differences with regards to performance and overall approach in the top-performing teams. One trend is the use of word and character n-grams as the most discriminating features, which is also commonly the case for the task of Native Language Identification (NLI) (Tetreault et al., 2017). In choosing a classification algorithm, four out of the five top teams opted to use linear support vector machines in their setup, which also echoes techniques used in NLI. The system paper describing the best submission (Bestgen, 2017) again underlines this methodological similarity, having drawn inspiration from entries in NLI shared tasks.

Alongside the main DSL task there were two specific tasks of language variety identification in 2017, namely Arabic Dialect Identification (ADI) and German Dialect Identification (GDI). The goal in both tasks is to identify a native language dialect in speech transcripts. For the Arabic variant, participants had access to transcripts and accompanying acoustic features from a multi-dialectal speech corpus and were tasked to discriminate between five Arabic dialects. A new dataset was developed specifically for the GDI task. The set contains transcribed interviews of one of four variants of Swiss German. The transcriptions use Schwyzertütschi Dialäktschrift which contains phonetic properties of language varieties (Tetreault et al., 2017).

The mentioned tasks on dialect identification are most closely related to the newly proposed DFS task, in that they deal with language variety, but the provided data in ADI and GDI is based on spoken data and provides some phonetic insight into these transcripts (either through separate acoustic Vectors or a rich phonetic transcript) whereas the DFS data does not.

We can, however, mention some promising approaches in the previous tasks using ensembles or meta-classifiers with a large variety of word and character n-gram features that yield top performances in ADI (Malmasi and Zampieri, 2017a) (second place) and GDI (Malmasi and Zampieri, 2017b) (first place). Malmasi and Zampieri use combinations of base classifiers to approach both tasks. In the first ensemble method, each base classifier votes for its most probable label output and the label with the most votes serves as the system's final output. The second ensemble method averages the probability outputs of all base classifiers and the class label with the highest averaged probability serves as the system's final output. The third variant, and most accurate approach, uses a Random Forest algorithm to meta-classify the probability distribution output of the base classifiers. The use of basic n-gram features with meta-methods seems to yield promising results regardless of the language varieties concerned.

The DFS data set and task are both closely related to and directly motivated by van der Lee and van den Bosch (2017), who offer the first comprehensive study of the automatic distinction between Dutch and Flemish. The authors explore lesser studied techniques, some of which apply specifically to the distinction between Netherlandic Dutch and Flemish Dutch. Syntactic features, of which part-of-speech n-grams yield the most important patterns, perform well even without combining them with word n-grams. Another important takeaway is that for their dataset, different feature groups are best combined using a meta-classifier. The configuration which combines all feature groups and trains a separate classification on their output probability distribution yields the highest F-score at 92 percent.

3 Data and methodology

In this section, we describe the characteristics of the provided data set and how this influences our methodology. We further select features and techniques that we deem important to try for the proposed task on the basis of the work discussed in the previous section.

Translation	Word form	Language variety	Occurrences in BEL	Occurrences in DUT
Onion	Ajuin	Flemish	6	0
	Ui	Netherlandic	46	53
Misery	Miserie	Flemish	12	0
	Ellende	Netherlandic	138	170
Microwave	Microgolf	Flemish	1	0
	Magnetron	Netherlandic	6	25
Cell phone	Gsm	Flemish	112	30
	Mobieltje	Netherlandic	15	51

Table 1: Excerpt of the lists of typical Flemish words and typical Dutch words with similar meanings. The frequencies of the words in the provided training set indicate the usefulness of the word list features over word n-grams because some words are very uncommon. This table somewhat overstates the differences between the Flemish and Netherlandic synonyms. For some words with similar meaning in the Wikipedia list there were no evident occurrence differences.

3.1 Data

The provided training data consists of 300,000 Dutch subtitles of which half were produced for Dutch television and half were produced for Flemish television (receiving the labels DUT and BEL respectively). A development set which contained 250 DUT-documents and 250 BEL-documents was also provided.

The subtitles originate from movies, television programs and documentaries but unlike the SUBTIEL corpus used in van der Lee and van den Bosch (2017), a single document does not represent as single movie or a single episode of a series. Rather, a document contains several (varying from one to thirty) lines and between thirty to 150 words.

We note two important implications from this prior data limitation: there is an expected performance drop compared to the work done by van der Lee and van den Bosch (2017) since each document contains fewer words, and some macro-features, notably document length features should not capture much in terms of language variety.

3.2 Features

We consider a top-down approach for our selection of features. Commonly stated language differences between Netherlandic Dutch and Flemish Dutch are in specific word choices (see Table 1) and word order choices. The latter is often a difference in placing finite verbs before versus after auxiliary verbs.

To see how word occurrence may differ we take an excerpt from the list of differences between Dutch and Flemish from Wikipedia (2018) and count occurrences of words in the training documents (Table 1). Using word n-grams with tf-idf weighting should let the classifier capture these and more subtle differences in occurrence patterns. We use unigrams up to trigrams to partly learn distinctive patterns of word order.

We also try lemma 1-3 grams. These features should be less important for capturing word order since removing inflection also means losing most morpho-syntactical information of the patterns. However, they can be useful for grouping words with diverse inflectional forms. We use Frog (van den Bosch et al., 2007) for finding the lemma of a given word.

Frog was also used to give detailed part-of-speech tags for our PoS n-gram features. We are interested in finding patterns of three or more words to capture distinctions in word type order between the target language varieties. Frog tags adhere to the CGN tagset (Van Eynde, 2004) and display a wide array of morpho-syntactic information. Verbs receive tags for their tense, whether it is singular or plural, finite or an infinitive, for example. We experimented with the level of detail in tagging to find the optimal configuration, but we found that more detailed tags consistently boost performance and pattern scarcity does not occur until we extend to 7-grams. Our final implementation relies on PoS patterns from trigrams up to 6-grams.

For our last group of features we were informed by work in Native Language Identification (Malmasi and Dras, 2017; Tetreault et al., 2017). Function words can feed important (and currently missing) stylometric information to our classifier. For our implementation of function word trigrams we remove words that are neither articles, pronouns, conjunctions or auxiliary verbs and construct patterns for the remaining words (regardless of the number of gap words). As with the other n-gram feature sets we apply tf-idf weighting on the word counts.

3.3 Classification methods

As a simplification we only consider the Linear Support Vector Machine (SVM) as a base classifier. The consideration is that SVMs have been shown to work well when using large feature sets and when making binary distinctions in the data. The algorithm is also fast to train and accurate in its default configuration, at least in its scikit-learn (Pedregosa et al., 2011) implementation, which further allows us to focus more on combining feature sets than finding optimal algorithms and parameters.

We consider three combination approaches:

1. **Default**. We combine the feature vectors and use only one SVM to find optimal separation in the feature space.

2. **Highest confidence vote**. Each of the feature vectors is fed into their own base classifier. Since each feature group is intended to capture a separate quality of the language varieties, which might be present or absent in the document, the classifier which outputs the highest probability for a certain label decides the final label.

3. **Meta-classifier**. Probability outputs of each of the base-classifiers serve as input for a meta-classifier. The meta-classifier should learn regularities in probability distribution such as in-balance between the labels and which of the feature sets makes less accurate predictions. We use Linear Discriminant Analysis as our meta-classifier here, which worked best in the case of NLI (Malmasi and Dras, 2017).

4 Results

Each of the feature groups are first tested separately on the development set to see their individual contribution to the classification results. We then test combinations of feature groups in each of the meta-classifier setups and motivate our selection of the three entries submitted to the shared task evaluation.

4.1 Feature group performance

Table 2 shows the F-scores of the individual feature groups. Although the linguistic information captured in the feature groups is different, performance of the feature groups is mostly similar. Only the function word n-grams perform very poorly with 55 percent F-score. We think this partly pertains to our operationalization of this feature set. It would be better to use stricter checks on what composes a function word, and to use placeholders to signal presence of other words. There can also be a limit on the number of words that may be skipped to form function word n-grams. Two function words that appear in separate sentences are not suited to be used as a bigram compared to two function words that have only one word between them. For further tests, we left out the function word n-grams because we expect them to harm performance in any configuration.

Feature group	N-gram range	F-score
Words	1-3	0.689
Lemmas	1-3	0.672
PoS	1-6	0.688
Function words	1-3	0.552

Table 2: Performance of individual features on the development set.

The content features (word n-grams) perform the best, but this is closely followed by the syntactic features (PoS n-grams). This again stresses the need for a suitable method to combine classifiers. Lemma n-grams perform slightly worse than word n-grams and since the feature groups capture somewhat similar language variety, it is important to consider if this feature group really contributes anything.

4.2 Combining feature groups

Table 3 compares the proposed classification combinations. A combination of the word and PoS n-grams features outperforms other combinations in the confidence vote and meta-classifier setting. No single classification combination consistently outperforms the others, which motivated us to make a submission for each of the proposed classification combinations.

Feature groups	**Classification combination**		
	Default	Confidence vote	Meta-classifier
Words + Lemmas	0.689	0.672	0.684
Words + PoS	0.693	**0.710**	**0.706**
Lemmas + PoS	0.680	0.694	0.684
All groups	**0.698**	0.697	0.692

Table 3: Combinations of feature groups using one of the three proposed methods. Meta-classification boosts performance in cases when the PoS feature group is included. The best combination of feature groups corresponds to individual feature group testing and our motivation to have features capture linguistic information. Confidence voting works best in this case, yielding 71 percent F-score on the development set.

4.3 Test set performance

The performances of our submissions show the same slight differences on the test data (see Table 4) as on the development data. Confidence voting again outperforms the others combination methods, followed by the meta-classifier and default vector combinations. Overall, the results on the test set are a lot lower than on the development set. There is no obvious way in which the test documents differ from those in the training or development set. The length of the documents is similar, and the distribution of the labels remained the same. In future attempts, it would be an improvement to evaluate the system during development using (embedded) cross-validation to better predict results on unseen data.

5 Discussion and Conclusion

In the following section we do a more in-depth discussion of the results and the different submissions. We first show which patterns in the features were captured by the system and why we believe they are distinctive for the two varieties of Dutch. We then critically discuss the submitted system and which improvements could be made.

5.1 Feature importance

Tables 5 and 6 show the most important features for both Flemish and Netherlandic Dutch by feature group as learnt by the base classifiers. The lexical differences contain mostly frequent interjections. This also explains why this table only shows unigrams.

Some unexpected results are ‘ele’, ‘enten’, ‘ent’ and ‘ine’ for Flemish Dutch. It turns out that these are suffixes that follow the special ‘ë’ or ‘ï’ characters. These characters were not processed correctly in the provided data, leaving a whitespace in their stead. Words containing these character and these suffixes such as ‘financiële’, ‘cliënten’, ‘patiënt’ or ‘cocaïne’ were more common in Flemish subtitles.

Submission no.	**Class. combination**	**F-score**
#1	Meta-classifier	0.629
#2	Confidence vote	**0.636**
#3	Default	0.627

Table 4: Test scores show the same pattern observed in the development set but our performance is consistently lower on the test set.

No.	BEL	DUT
1.	ele	oke.
2.	enten	oke,
3.	da‘s	he?
4.	ent	masterchef
5.	allee,	text
6.	komaan,	he.
7.	ine	grayson.
8.	vanop	eh,
9.	sami	eh.
10.	amai,	he,

Table 5: Most important word n-gram features. We find expected values and some that are particular to the subtitle data set. Future work could use intensive preprocessing to limit observed artifacts such as misformatting and genre-specific information.

No.	BEL	DUT
1.	IN NP SYM	VVZ VV PP
2.	NP PP$ N	VVD VVD PP
3.	NP SYM SENT	PP(‘Er‘) PP(fin) SENT
4.	SENT NP PP$	VV PP(‘ie‘) SENT
5.	SENT PP(‘U‘) IN	VVD VVN VVD
6.	IN NP PP$	FW SENT SENT
7.	PP$ NNS SENT	IN PP SENT
8.	SENT SYM(‘Da’s‘) JJ	UH(‘O‘) UH(‘Ja‘)
9.	IN NP SYM SENT	RB(‘Zo‘) UH(‘Ja‘) SENT
10.	IN PP(‘dat‘) SENT	NN NN VV

Table 6: Most important part-of-speech n-gram features. Since we start with trigrams, these are more informative for analysing specific word orders. They are expressed in the Penn Treebank tagset to promote reader comprehensibility, but direct translations were not always possible. In cases where a tag referred to mostly one specific word, this word is included in brackets. We find new distinctive linguistic patterns that were not expected a priori.

‘Sami’ for Flemish and ‘Masterchef’ and ‘Grayson’ are also surprising terms in the list of top most distinctive features. These artifacts of the training data indicate that some television content was more common for one of the language varieties. For any shared task such artifacts pose a dilemma as removing them, for example through named entity recognition, would be more faithful to the task of language variety identification, but could potentially harm the competitive performance. We decided to use any information available in the subtitles to train a competitive system.

The part-of-speech n-grams (Table 6) are also interesting to analyze and contain some new patterns that we did not expect a priori. In section 3.2 we mentioned the order of finite and auxiliary verbs but we do not find this as one of the most important features. Instead, a common pattern in Flemish Dutch (in features 2, 4, 6 and 7) is using possessive pronouns after proper nouns whereas Netherlandic Dutch tends to use a possessive ‘s’ in such cases. This was also found in van der Lee and van den Bosch (2017). Examples from the training set include ‘Coryn haar moeder’ and ‘Tom zijn karakter’ for Flemish versus ‘Henry‘s auto’ and ‘Anne‘s verdwijning’ for Netherlandic.

Our findings also echo some lexical characteristics for Flemish. The specific proper noun ‘U’ is more common, as well as two-word contraction ‘Da’s’ which is tagged as a special ‘incomplete’ sign in our tagset (Van Eynde, 2004). The Netherlandic features show similar findings. Multiword interjections such as ‘Oh ja’, ‘Zo ja’ and ‘Maar ja’ are strong indicators for the label, but are sometimes mistagged. The personal pronoun ‘ie’ is indicative for Netherlandic as is ‘er’.

On a syntactic level we find an interesting use of verbs in features 1, 2 and 5 for Netherlandic Dutch. It is not uncommon to see three consecutive verbs such as in the training sentence:

“Na alle verderf die ik had veroorzaakt was Lana mijn hoop op verlossing.”
(‘After all the misery I had caused her, Lana was my hope for salvation.’)

In Flemish Dutch, commas are regularly placed before the last verb making this construction less common. This may be part or all of the reason why van der Lee and van den Bosch (2017) find the use of commas to be indicative of Flemish Dutch. An example training sentence is:

“De priester die me had opgeleid, was teleurgesteld.”
(‘The priest who trained me was dissappointed.’).

The syntactic patterns we extracted from just the top ten most distinctive features show interesting differences between Flemish Dutch and Netherlandic Dutch. In future work these differences could be explicitely modelled for the task of NLI. We generally see that the tags that we used and the range of our n-grams are sometimes too detailed to capture the differences we found upon later inspection. Future work could benefit from an iterative approach to its feature design.

5.2 Conclusion

Our second submission, which implemented the confidence vote combination of classifiers, yielded the best results. Overall it achieved third place out of twelve at the DFS shared task. The first place submission achieved an f-score of 66 percent, which is considerably higher than our own result.

In this particular task setup, we think the confidence vote outperformed the meta-classifier because clues of language variety can vary greatly between the short documents in the data set. One document may contain a syntactic pattern typical of Flemish Dutch whilst containing words that are predominantly used in Netherlandic Dutch. A meta-classifier can learn that the probabilities that are output by the word n-gram classifier are generally more accurate than those output by the PoS n-gram classifier. The confidence vote approach allows this general logic to be overruled in cases that are easier to predict for one of either classifier; the n-gram classifier is ignored in documents with distinctive syntactic patterns.

Our n-gram feature groups performed well on this task but could further be improved. In any n-gram configuration, performance can be boosted by leaving very common or very uncommon patterns out of the feature. Very common uni and bigrams may not be informative and needlessly influence the classification while very uncommon patterns are hard to generalize and may lead to overfitting on the training set.

The part-of-speech n-grams were a useful feature group for discriminating Flemish Dutch from Netherlandic Dutch. The level of detail in the part-of-speech tags is something to critically analyze in future work. As we showed with a manual inspection of the most import PoS-patterns, the important differences between the language varieties on a syntactic level did not require as much detail in the tags to be captured. Because of the specificity of the tags, some of the found patterns also overlap with the lexical features. The pattern 'SENT SYM(incomplete) JJ' (PoS feature eight for Flemish) almost always refers to 'Da's' which was already captured using the word n-grams.

The function words for modelling stylometric aspects did not work with our operationalization. As noted, our implementation which skipped over arbitrarily long sequences of words to form function word n-grams was not optimal. It can be interesting to use function word co-occurrence as a feature, but different distances between these words should not be treated as the same pattern. We suggest the use of placeholders to signal the number of skips, or to limit the length of the skipped sequence.

Given time we would have used cross-validation. It would have given us more accurate performance assessments during development and would have allowed optimization of hyperparameters.

We only tried Linear Discriminant Analysis for the meta-classifier due to time constraints, but other options are available and might have performed better. Our result, which shows confidence voting outperforming the meta-classifier consistently should always be seen in the context of this specific task and with the specific meta-classifier we used. There are several other ways to combine classifiers using some form of averaging or voting that can also be explored.

In conclusion, we used a very directed approach to test features based on linguistic knowledge of both language varieties and found that these features added important discriminating information to the system. We also tried several classifier combinations, and found that a confidence voting algorithm outperformed a meta classifier in this specific task.

This project further yields a number of directions for future work to be explored for similar tasks or for providing a more extensive analysis of the differences between Dutch language varieties. We encourage such work to use our system, which is available from the CLiPS GitHub [1].

[1] https://github.com/clips/vardial-dfs

References

Yves Bestgen. 2017. Improving the character ngram model for the dsl task with bm25 weighting and less frequently used feature sets. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 115–123.

Cyril Goutte, Serge Léger, Shervin Malmasi, and Marcos Zampieri. 2016. Discriminating similar languages: Evaluations and explorations. *arXiv preprint arXiv:1610.00031*.

Shervin Malmasi and Mark Dras. 2017. Native language identification using stacked generalization. *arXiv preprint arXiv:1703.06541*.

Shervin Malmasi and Marcos Zampieri. 2017a. Arabic dialect identification using ivectors and asr transcripts. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 178–183.

Shervin Malmasi and Marcos Zampieri. 2017b. German dialect identification in interview transcriptions. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 164–169.

Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Joel Tetreault, Jill Burstein, Claudia Leacock, and Helen Yannakoudakis. 2017. Proceedings of the 12th workshop on innovative use of nlp for building educational applications. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*.

Antal van den Bosch, Bertjan Busser, Sander Canisius, and Walter Daelemans. 2007. An efficient memory-based morphosyntactic tagger and parser for dutch. *LOT Occasional Series*, 7:191–206.

Chris van der Lee and Antal van den Bosch. 2017. Exploring lexical and syntactic features for language variety identification. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 190–199, 4.

Frank Van Eynde. 2004. Part of speech tagging en lemmatisering van het corpus gesproken nederlands. *KU Leuven*.

Wikipedia. 2018. Lijst van verschillen tussen het nederlands in nederland, suriname en vlaanderen — Wikipedia, the free encyclopedia. [Online; accessed 03-May-2018].

Marcos Zampieri, Shervin Malmasi, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, Jörg Tiedemann, Yves Scherrer, and Noëmi Aepli. 2017. Findings of the vardial evaluation campaign 2017. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 1–15, Valencia, Spain, April. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shon, James Glass, Yves Scherrer, Tanja Samardžić, Nikola Ljubešić, Jörg Tiedemann, Chris van der Lee, Stefan Grondelaers, Nelleke Oostdijk, Antal van den Bosch, Ritesh Kumar, Bornini Lahiri, and Mayank Jain. 2018. Language Identification and Morphosyntactic Tagging: The Second VarDial Evaluation Campaign. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, Santa Fe, USA.