Natural Language Engineering (2019), 1–18
doi:10.1017/S1351324919000445

CAMBRIDGE UNIVERSITY PRESS

ARTICLE

Domain bias in distinguishing Flemish and Dutch subtitles

Hans van Halteren*

Centre for Language Studies, Radboud University, Nijmegen, The Netherlands
*Corresponding author. Email: hvh@let.ru.nl

(Received 17 October 2018; revised 20 June 2019; accepted 26 June 2019)

Abstract

This paper describes experiments in which I tried to distinguish between Flemish and Netherlandic Dutch subtitles, as originally proposed in the VarDial 2018 Dutch–Flemish Subtitle task. However, rather than using all data as a monolithic block, I divided them into two non-overlapping domains and then investigated how the relation between training and test domains influences the recognition quality. I show that the best estimate of the level of recognizability of the language varieties is derived when training on one domain and testing on another. Apart from the quantitative results, I also present a qualitative analysis, by investigating in detail the most distinguishing features in the various scenarios. Here too, it is with the out-of-domain recognition that some genuine differences between Flemish and Netherlandic Dutch can be found.

Keywords: Text classification; Dutch; Methodology; Dialect recognition; Topic bias

1. Introduction

For many tasks relating to, a.o., natural language processing or language variation studies, one of the first steps is to determine which language, language variety or dialect a text is written in, that is, Language Identification (LI). It therefore does not come as a surprise that LI is well established, with computational work having started more than 50 years ago (Mustonen 1965). It is outside the scope of this paper to present too many details of work in the field, but I refer the reader to Jauhiainen *et al.* (2018) for a recent and excellent overview. Lately, LI research is aimed at the more challenging aspects, such as LI in documents with code switching, for example, pursued in the series of Workshops on Computational Approaches to Code Switching with shared tasks (Solorio *et al.* 2014; Molina *et al.* 2016; Aguilar *et al.* 2018), and distinguishing between very similar languages, or varieties and/or dialects of the same language, as for example, pursued in the VarDial/DSL series of workshops and shared tasks (Zampieri *et al.* 2014; Zampieri *et al.* 2015; Malmasi *et al.* 2016; Zampieri *et al.* 2017; Zampieri *et al.* 2018).

A problem with these tasks, and in fact with many text classification tasks, is the construction of an experimental data set in which the texts of the various classes only differ as to their assigned class, and not in any other variable. Very often, quality measurements are confounded by one or more other variables and authors rightfully complain about some appropriately named bias, for example, *topic bias* (Malmasi *et al.* 2015).[a]

[a]Beware that such a bias comes in two forms. Very often the situation addressed is one where test data are of a different nature than training data, and that models perform badly because of a bias towards the training data. In the second form, classification performs better than expected because separate classes are biased differently and the classifier (also) picks up clues on the bias. This paper focuses on the second form.

A bias was also present in the Dutch–Flemish Subtitle (DFS) task of VarDial2018 (Zampieri *et al.* 2018), where I participated together with Nelleke Oostdijk (as Team Taurus). Given Dutch subtitles produced in the Dutch and Flemish branches of Broadcast Text International (BTI), the task was to recognize which variety of Dutch[b] was represented by an item, consisting of around 30 words. Even though the two varieties should be almost identical, the best system (Çöltekin *et al.* 2018) reached an F1-score of 0.6600, closely followed by our submission (0.6456). A possible explanation for this relatively high score was the wealth of data: 150,000 training items for each variety, far more than for the other VarDial2018 tasks. This suggested explanation might be supported by the results reported by van der Lee and van den Bosch (2017), who claimed an F1-score of 0.92 when distinguishing between the varieties on the full Subtiel corpus (150 million tokens).[c] However, when we attempted to identify differences between Flemish and Netherlandic Dutch on the basis of the VarDial2018 data (van Halteren and Oostdijk 2018), we found that most of the recognition quality seemed due to recognition of specific movies and/or TV shows (henceforth 'programs'). It turned out that both training and test material had been sampled randomly from the whole data set, so that test items might be from the same program as training items, and be recognizable by source-specific words, such as proper names.

In this paper, I investigate how strong the interference of domain information is in this task, using data from the Subtiel Corpus (van der Lee 2017; van der Lee and van den Bosch 2017) that was used for VarDial2018 DFS, but then with attached metadata.[d] From these data, I clustered the items on the basis of content word similarity of programs and used the clusters to create two sets of items, each containing one million words of both Flemish and Netherlandic items, internally without overlap in program or in content cluster. I then repeated the Vardial2018 DFS task, both with within-set train-test splits and between-set train-test splits. In this way, I attempted to maximize and minimize domain interference. The degree of interference became visible both in the difference in recognition quality and in the most distinguishing features.

After discussing some related work (Section 2), I introduce the extended data used in this paper and the processing leading to features for language variety recognition (Section 3). Then Section 4 describes the actual recognition process and its quantitative evaluation. I follow this up with a more qualitative evaluation in Section 5, investigating the most distinctive unigrams in various scenarios. After this I conclude (Section 6) with a discussion of the most important observations and what they imply.

2. Related work

Whenever data sets are created without specific attention to biases, for example, when there is a sparseness of data or an existing corpus is being reused, biases can be expected to turn up. This is not different in language variety/dialect identification, the background task in this paper. And unless the task is specifically aimed at the bias itself, it tends to be discovered only after the main recognition task has been performed. Sometimes, the researchers find hard to explain differences between tests on different subsets of the data. As an example, Hanani *et al.* (2017) mentioned topic bias as a likely factor of system behaviour on a Swiss–German dialect recognition task, compared to that on Arabic dialect recognition. After suggesting the influence of bias, they give examples of features which are topical in nature rather than dialectal, such as the tokens 'Zürich' and 'Berne'.

[b]The two classes are indeed varieties and not dialects, as the subtitles in question are supposed to be written in both countries in standard (written) Dutch. Differences should therefore be minimal. Still given the development of the use of Dutch as described by, for example, Grondelaers and van Hout (2011) and De Caluwe (2012), differences should not be completely absent either.

[c]However, in their experiments the test items were also much larger, namely documents with all subtitles for a full movie or TV show episode.

[d]I am very grateful to Chris van der Lee, who managed to link many VarDial DFS items to the original metadata in the Subtiel Corpus and shared these data with me.

The term *topic bias* in dialect recognition was introduced by Malmasi *et al.* (2015).[e] They paid specific attention to the bias in Arabic dialect recognition, and used mostly balanced material for their main experiments. In addition, they tried to measure the impact of the bias by cross-corpus classification experiments. They concluded that 'a key finding here is that the models trained here do generalize across data sets with a high degree of accuracy, despite their striking differences in size and content. Although this result does not evidence the absence of topic bias, it may indicate that its negative effects are tolerable', p. 207.

As the literature on biases in dialect recognition is as yet limited, I turn to authorship studies, where the influence of topic, genre and medium is well known and well discussed. As this field pays more attention to the actual features used in recognition, the presence of biases is hard to miss, witnessing, for example, Tables 5 and 6 in the study by Koppel *et al.* (2009), which show that for gender, age and mother tongue classification content features perform better than style features, and which list examples of high scoring features like 'boyfriend' (female), 'software' (male), 'school' (teens), 'apartment' (twenties), 'wife' (thirties+), 'bulgaria' (Bulgarian) and 'russia' (Russian). The same overall picture is painted by van Halteren and Speerstra (2014), where gender recognition on Dutch tweets is carried almost completely by content words related to the topics discussed by (young) males and females on Twitter. Perusal of overview papers, such as that by Stamatatos (2009), shows various cautionary remarks about topic and genre, but furthermore gender, age, education level, nationality, etc., namely all factors that might influence language use other than the identity of the author.

Two main solutions are being proposed. The first is using balanced corpora, controlled for as many interfering factors as possible. An example is the ABC-NL1 corpus compiled by Baayen *et al.* (2002), comprising nine texts each by eight Dutch students of Dutch. The nine texts were prompted and there were three texts each in the argumentative, non-fiction and fiction genre, with vastly differing topics. The spread in genre/topic and the symmetry of the design made this corpus ideal for author recognition, as can be seen from the potential of almost perfect recognition (van Halteren 2007). However, it is obvious that such a design is only useful for in-vitro testing of recognition methods. The probability of coming across such an ideal corpus for any practical tasks is slim indeed; Chaski (2001; esp. Appendix 1) compiled a more task-oriented balanced corpus (the task being forensic linguistics), the Writing Sample Database. The database consists of 10 prompted texts by 92 authors and metadata on various sociolinguistic parameters, allowing the selection of balanced subsets. Attempts to build larger controlled corpora have been made, for example, the Usenet Post Corpus by Argamon *et al.* (2003), but control here is often limited to topic, as such scraped corpora tend to lack metadata on the authors.

A second solution is the attempt to identify which recognition features are more related to authorship and which more to the bias (and which to both). Here I refer the reader especially to Mikros and Argiri (2007), who, after an ample introduction to the topic bias problem, discuss experiments with a specially compiled corpus, controlled for author and topic, and consisting of 200 articles written by two authors (with a similar writing style) for two major Greek newspapers on the topics of Culture and Politics. With this corpus, they tested a large number of commonly used stylometric features, using two-way ANOVA to measure how related each feature was to the individual factors, author and topic, and to their interaction.[f] They conclude that the tests 'revealed that many stylometric variables are actually discriminating topic rather than author. Among them, we found Frequent Function Words, specific characters, word lengths and commonly used lexical richness measures, such as Yule's K'. They advise extreme caution in applying these stylometric variables for authorship attribution on multi-topic corpora, in order to avoid

[e] In this paper, I use *domain bias* rather than *topic bias*, as not all biased features are topical, but do all have a relation to the domain of program subtitles.

[f] In cases of non-normally distributed features, a Mann–Whitney U-test confirmed the ANOVA results.

that 'authorship attribution could become a by-product of the correlation of authors with specific topics'.[g]

Comparing the two fields, the 'extreme caution' for authorship recognition (Mikros and Argiri 2007) appears to conflict with an indication 'that the negative effects might be tolerable' (Malmasi *et al.* 2015) for dialect recognition. As the tolerance may be related to the language and feature types in question, I aim to examine the (potential) problem with a vastly different language and feature set in this paper.

3. Data

3.1 DFS data with metadata

For VarDial2018 DFS, items were distributed without any metadata, that is, only their text and a marker BEL (for Flemish) or DUT (for Netherlandic). For the current investigation, Chris van der Lee tried to establish links between metadata as present in the original Subtiel Corpus and the VarDial2018 DFS items. For all but 97 items, a link to the original corpus was found. However, for 135,253 items (42.2%) there was no metadata present in the corpus. For the remaining 185,150 items, I received access to the program name and an indication of the genre, for example, 'Action, Crime, Drama' or 'Reality-TV'. The latter did not appear very consistent at first sight and certainly not strictly correlated with my concept of domain, so that I only used the program name. I would have liked also to investigate any influence of author idiolect, but metadata on authorship was unfortunately not present.

3.2 Domain separation

For my new experiments, I wanted to separate the items into two sets, between which the domain overlap would be minimized. Applying a train-test split at program level is insufficient, as various programs will undoubtedly be similar. Clear examples are Poker After Dark, Late Night Poker and Poker Nations Cup, or the multitude of cooking shows in the corpus. I therefore decided to attempt to cluster the data at the program level, using content word similarity for purposes of clustering. In order to have a sufficient amount of text for each program, I only used programs for which there were at least 1000 words. There turned out to be 918 such programs, with a total of 5.3 million words.

The next step was to build vectors of word counts for these 918 programs. For this, words were taken to be sequences of letters (cast to lower case) and digits, limited to those which occurred at least three times in the corpus.[h] I weighted the counts in the vectors with Inverse Document Frequency (IDF) at the program level, using the 918 selected programs, and took cosines between each pair of vectors. I then built a full network between the programs, with these cosines as weights, and applied the Weighted Louvain Algorithm for community detection in large networks (Blondel *et al.* 2008; Campigotto *et al.* 2014).[i]

Application of (the weighted variant of) the algorithm on the set of 918 programs yielded 4 communities at the second (and last considered) level. Their main characteristics are shown in

[g]With the noted exception that there might be situations where putting the bias to good use is appropriate for the task at hand, for example, creating large test corpora in the first place.

[h]In the recognition experiments below, I worked with preprocessed and more properly tokenized text, which was there needed to derive some of the feature types. For the train-test splits, such processing is not needed and avoiding it also avoided risks of erroneous analyses.

[i]Code provided at https://sourceforge.net/projects/louvain/. As Blondel *et al.* (2008; Section 2) explains, the algorithm attempts to find 'high modularity partitions' in large networks. It first builds communities of individual network nodes, in such a way that the modularity is locally maximized. The communities are then taken to be nodes in a new network, and the process is repeated. They state that iteration continues 'until there are no more changes and a maximum of modularity is attained', at which point a 'complete hierarchical community structure' has been determined.

Table 1. Program 'communities'

Community	Programs	Words Flemish	Words Netherlandic	Dominated by
1	339	1,343,512	1,010,084	Soaps, sitcoms, talkshows, talent shows
2	254	453,886	742,888	Cooking, sports, dancing, games
3	28	85,019	119,621	Hospital series
4	295	698,057	853,644	Crime, action, scifi-fantasy

```
s/\([Gg]e\) dealiseerde /\1idealiseerde /g
s/i re /iere /g
s/\([0-9]\)\([,.]\) *\([0-9]\)/\1\2\3/g
```

Figure 1. Examples of regular expression substitutions for normalization.

Table 1. As the algorithm chose not to do another iteration, I assumed that the final communities did not show much overlap in content anymore, so that I could freely combine them into two groups on the basis of size criteria alone. The optimal combination was to put communities 1 and 3 in group A and communities 2 and 4 in group B. In this way, each group provided at least one million words on both the Flemish and Netherlandic side, from which I then selected exactly one million words for each block (BelA, BelB, DutA, DutB) by random selection at the item level.[j]

Having selected my experimental data using the original VarDial format, I proceeded with normalization and analysis as applied during the VarDial competition. To be exact, I reused the features extracted for VarDial, selecting only the items in my new data set. As will become clear in the following paragraph, these features are not optimal, but the decision was taken to remain comparable with the VarDial experiments for the current and future papers.

3.3 Text normalization

In our[k] processing of the text for VarDial, the first step was normalization. The reason for this was that the text as present in the VarDial items contained several artefacts of earlier preprocessing steps. Most notably, all characters with diacritics were removed (e.g. *één* ('a') became *n*), or alternatively the diacritic was removed but also a space was inserted (e.g. *ruïne* ('ruin') became *ru ine*). Furthermore, periods in numbers had spaces inserted next to them (e.g. *20.000* ('20,000') became *20. 000*). Also apostrophes in words like *z'n* (*zijn*, 'his') were removed. Apparently, some correction had already been applied, but this also produced non-existing forms like *zeen*.

Such artefacts would not be a problem for character or token *n*-gram recognition. However, we wanted to apply Part-of-Speech (POS) tagging and syntactic parsing, for which these artefacts would most certainly lead to errors. We therefore decided to include a preprocessing step in which we tried to correct most of these artefacts. For the diacritics, it would have been easiest to compare to a Dutch word list in which diacritics were included, but we did not manage to acquire such a resource in time. Instead, we inspected derived word counts and the text itself manually, and built a list of about 270 regular expression substitutes, such as the ones shown in Figure 1. As we spent only limited time on this, we missed cases, even (in retrospect) obvious ones like *financi ele* (*financiële*, 'financial'), leading to non-words like *ele* in the observations below. As we reused

[j] A full list of programs in each community can be found at https://cls.ru.nl/staff/hvhalteren/vardial_revisited/communities.txt and the list of VarDial 2018 DFS items selected for the four one million words samples at https://cls.ru.nl/staff/hvhalteren/vardial_revisited/blockitems.txt.

[k] This section describes work done together with Nelleke Oostdijk for VarDial2018.

```
Voor     VZ(init)
vandaag  BW()
had      WW(pv,verl,ev)
ik       VNW(pers,pron,nomin,vol,1,ev)
al       BW()
een      LID(onbep,stan,agr)
maand    N(soort,ev,basis,zijd,stan)
geen     VNW(onbep,det,stan,prenom,zonder,agr)
stem     N(soort,ev,basis,zijd,stan)
meer     VNW(onbep,grad,stan,vrij,zonder,comp)
gekregen WW(vd,vrij,zonder)
.        LET()
```

Figure 2. Example: POS tagging.

```
<NOFUN>:SMAIN(krijg) -> [ mod auxv su mod obj1 mod lexv <NOFUN> ]
     mod:PP(voor|vandaag) -> [ hd obj1 ]
          hd:VZ(voor) -> voor
          obj1:BW(vandaag) -> vandaag
     auxv:WWaux(heb) -> heb
     su:VNW(ik) -> ik
     mod:NP(maand) -> [ mod det hd ]
          mod:BW(al) -> al
          det:LID(een) -> een
          hd:N(maand) -> maand
     obj1:NP(stem) -> [ det hd ]
          det:VNW(geen) -> geen
          hd:N(stem) -> stem
     mod:VNW(meer) -> meer
     lexv:WWlex(krijg) -> krijg
     <NOFUN>:.(.) -> .
```

Figure 3. Example: syntactic analysis.

the VarDial features, these missed cases have not been corrected, as mentioned for reasons of comparability.

3.4 Syntactic annotation

The normalized text was subsequently analysed by a combination of Alpino[l] and Frog[m]. An example of the POS tagging thus derived can be seen in Figure 2. We used the POS tags by themselves, but also less fine-grained POS classes (by discarding all but the first attributes of the various tags), leading to POS group tags like *WW(inf)* and *N(soort)*. Also, apart from the POS tags, the analysis provided us with lemmas, or rather stems as for example, for *gekregen* ('received'), we find the first person singular *krijg* ('receive').

Among other things, the Alpino/Frog combination yields a syntactic dependency analysis. However, as the dependency structure is less amenable to variation studies than a constituency structure, we first transformed the trees. We started with the 'surfacing' procedure developed by Erwin Komen (2015), which attempts to extract a constituency parse from the Alpino dependency parse and followed it up with a few more transformations, especially around the verb phrase. Furthermore, the analyses were lexicalized by percolating the head words upwards. As an example the parse of the sentence in Figure 2 is shown in Figure 3.

[l]A hybrid syntactic dependency parser for Dutch with a expert-written grammar-based core and a probabilistic final selection component (Bouma *et al.* 2001).

[m]A machine-learning-based system producing various linguistic annotations for Dutch, among which POS-tagging and syntactic analysis (van den Bosch *et al.* 2007).

Table 2. Feature types

Feature type	abs/lex	Example	Exam pref	Number
Char *n*-grams	lex	C2_!#	Dut	295,943
Token 1-grams	lex	T1_W_Komaan	Bel	64,051
Token 1-grams	abs	T1_P_VNW(pers,pron,obl,nadr,3p,mv)	Bel	246
Token 2/3-grams	lex	T2_WG_Oke_LET()	Dut	10,307,582
Token 2/3-grams	abs	T3_GPP_N(soort)_VZ(init)_ N(soort,ev,basis,dat)	Dut	260,850
Syn *n*-grams	lex	SCFFCCL_PP_hd_VZ(naar)_obj1_ VNW(hier)	Bel	384,185
Syn *n*-grams	abs	SCFFCC_CONJ_cnj_TW_cnj_LID	Dut	159,918
Syn rewrites	abs	SRFC_SMAIN_su_N_auxv_WWaux_ mod_BW_lexv_WWlex_<NOFUN>_.	Bel	40,210

3.5 Feature extraction

From the analysis results, we extracted various features which for this paper are represented as divided into eight types (Table 2).

The first type is formed by the character *n*-grams, with *n* ranging from 1 to 5. *n*-Grams (also token *n*-grams) at the beginning and end of sentence were facilitated by inserting hash characters (#) before and after each sentence.

Then there are four types of token *n*-grams. From the POS tagging, we derived unigrams, bigrams and trigrams. In each of the positions of the *n*-grams, we put one of the following: the word (W), the lemma (L), the full POS tag (P) or the POS group tag (G). As an example, T3_GLP_LID(bep)_ding_LID(bep,gen,evmo) is the trigram built with a POS group tag, a lemma and a full POS tag, that can, for instance, be found in the text fragment *de dingen des* ('the things of'). In this paper, the token *n*-gram features are divided along two dimensions. The first concerns the size of *n*: the unigrams are taken separately and the bi- and trigrams are kept together. The second concerns the use of lexical information. In the abstract *n*-grams (abs), the features do not contain any references to words or lemmas, so only full POS tags and/or POS group tags. In the lexical *n*-grams (lex), at least one field is a word or lemma.

From the syntactic constituency trees, we derived three types of features. Two of these contain a kind of syntactic *n*-grams[n] built by taking subtrees such as a functional constituent (F) realized by syntactic category (C) containing a functional constituent realized by syntactic category (e.g. SFCFC_mod_WHREL_obj1_TW, a modifier realized by a *wh*-relative clause (WHREL) containing a direct object (obj1) realized by a cardinal numeral (TW); or SCFFCCL_NP_hd_N(ding)_mod_NP(leven), a noun phrase (NP) containing both a head (hd) realized by a noun (N) with lemma *ding* and a modifier (mod) realized by a noun phrase (NP) with a head *leven*, which can, for example, be found for the phrase *de dingen des levens* ('the things of life')). These syntactic *n*-grams are again split into lexical (lex) and abstract (abs), depending on whether they refer to words or lemmas. The final type of syntactic feature comprises the full rewrites at all positions in the tree, for example, SRFC_WHQ_whd_VNW_hd_WWlex_obj1_TW_<NOFUN>_. (a *wh*-question realized by a interrogative pronoun, a verb and a direct object realized by a cardinal numeral, ending with a sentence closing punctuation mark).

[n] van der Lee and van den Bosch (2017) also use 'syntactic *n*-grams', but they use this term for *n*-grams of POS tags, whereas we take subtrees of a syntactic analysis tree.

4. Quantitative analysis

4.1 Recognition procedure

As during VarDial 2018, my choice of recognition system was based on the goal of finding differences between the two language varieties. After successful recognition, I wanted to be able to identify which features contributed to the success. I therefore used a very simple algorithm. I simply counted the occurrences of each feature in the Flemish and Netherlandic training items and compared the two counts to derive odds.[o] For example, the word bigram *Komaan_,* ('Come on',) was found 209 times in the Flemish items and four times in the Netherlandic items, leading to odds of 52.25:1 in favour of Flemish Dutch. If one of the counts was zero, it was counted as 0.5 for determining odds.[p,q]

During testing, the algorithm examined each feature for a test item. If its odds were equal or higher than 2:1, it gave points to the favoured language, equal to the odds. Odds in favour of Flemish are represented as positive numbers, odds in favour of Netherlandic as negative ones. The sign of the sum over all (activated) features determined the final choice. If, due to absence of any distinguishing features or the coincidence of total balance, the final result was zero, a random choice was made (as the original task was defined as a forced choice).

In principle, this is a marker-based approach. I am searching each item for clear markers for either variety. This is why I only use features with odds at least 2:1.[r] This approach is appropriate for items consisting of so little text, too short for applying full knowledge about under- as well as overuse. The idea of markers also explains why I am adding odds rather than multiplying, which would be more natural mathematically: with addition, the stronger markers are given more impact than with multiplication. Having said that, in the experiments in this paper I attempted both addition and multiplication, and found addition to be slightly better most often, but not significantly so. The results presented below are those with addition.

For this paper, I remained true to the recognition approach used in VarDial. I expect that the major effects shown below will occur with any learning method and my chosen one just makes it easier to identify distinguishing features. There was, however, one major change. During VarDial, we used add-one smoothing in cases where the count for the underrepresented language variety was zero. For these new experiments, I first investigated Good-Turing smoothing on all feature counts, but found the feature frequency distribution to be not really amenable to this method.[s] In the end, I chose to use 0.5 for the odds calculation for features not seen on one side, as was already mentioned above.[t,u]

[o] Using absolute counts is possible as all data blocks are of the same size. In other cases, relative frequencies would be needed, in which cases the smoothing procedure (see below) becomes more complicated. Furthermore, leave-one-out testing, as used below for some situations, would be almost impossible.

[p] See below for reasoning behind this.

[q] The odds method used here does not fit into the classification of methods by Jauhiainen *et al.* (2018), as it is geared specifically to a choice between two options, whereas most formulae presented by Jauhiainen *et al.* (2018) are geared to choosing between many languages at once. Feature values based on a comparison of frequencies in two classes do exist (e.g. Murthy and Kumar (2006)), but tend to be then inserted in more intricate algorithms than simple addition.

[r] This threshold was chosen intuitively.

[s] Good-Turing expects all counts to belong to one probability distribution, for example, those of all words. With our feature set, this is not the case, for example, the syntactic features are not all mutually exclusive. Good-Turing would be applicable to some subsets of features, but for reasons of comparability I did not want to use different types of smoothing for different features.

[t] 0.5 too was chosen intuitively, it being in between the zero leading to computational problems and one being the lowest value actually observed.

[u] Apart from those mentioned (odds threshold, calculation of total score, smoothing method, assigned value for unobserved features), there are no hyperparameters that need tuning. For these four, I chose intuitively and/or after a minimal pilot experiment. It may well be that better results can be derived with tuning of these hyperparameters, and certainly with more advanced classification methods, but that is not the point of this study.

4.2 Evaluation procedure

The focus of this paper is on the influence of the domain on language variety recognition. This means that I needed to vary domain choices in training and testing. As 'domains' I used the program groups A and B as identified above. Below I identify the various block in my data as 'BelA', 'BelB', 'DutA' and 'DutB', combining the VarDial DFS variety markers BEL (Flemish) and DUT (Netherlandic) with the domain markers A and B.

The situation during VarDial 2018 DFS itself was that training and test material was from the same domain and test items could be from programs also present in the training material. With my new data, this corresponded to both training and testing on BelA together with DutA, or on BelB together with DutB. In the following, I refer to this situation as 'in-domain'.

The situation in which domain influence is reduced the most is that in which I train on BelA and DutA, and then test on BelB and DutB, or vice versa. This I refer to as 'out-of-domain'.

I added two more situations to demonstrate the most extreme influence. In the first one, referred to as 'aided', I chose items in which the variety choice correlated with the domain choice; for example, I trained on BelA and DutB and also tested on BelA and DutB. If an item was correctly recognized, this may have been caused by either domain or variety recognition (and sometimes both). In the other situation, referred to as 'hindered', I used the same training selection, but tested on the unrepresented items, for example, I trained on BelA and DutB but tested on BelB and DutA. This meant that language variety recognition had to overcome the interfering signals stemming from domain differences.

For all combinations of training and test data, I used the same evaluation setup. In VarDial, there was an absolute train-test split (150,000 train items, 250 development and 10,000 test items for each variety). This was a necessity in the shared task scenario, and possible with the luxurious amount of data that was present. For my current experiments, this luxury was no longer present (around 30,000 items per block) but neither was the necessity. I could therefore use some kind of cross-validation. Here, my chosen recognition method turned out to have an additional advantage. If the tested block was also already used in training, I could still apply leave-one-out testing. As I kept all original counts for each block, I only needed to subtract the count contributions of a specific item when testing on that item. For all cases where the test item was not involved in training, no such measures were needed.

4.3 Recognition quality

I measured the recognition quality with *accuracy* (the fraction of test items for which the system chooses the correct variety) on each individual block. In VarDial, the ultimate measure was averaged F1-score, but in case of a forced choice between a finite number of options (here two) for a fixed item set, precision and recall (and then also F1-score) are all equal, namely equal to the accuracy. This was convenient now, as in each block only one language variety was represented so that precision became meaningless, but the accuracy was still comparable to the F1-score in VarDial.

Table 3 shows the accuracies for all pairings of train and test combinations. As could be expected, the aided recognition (in bold face type) worked best by far. Domain recognition was clearly helping a lot here (example features are discussed in the following). The strength of domain recognition becomes even clearer for the hindered recognition (underlined), which was pushed far under random selection (which would have an accuracy of 0.5); in other words, domain differences outweigh variety differences when using all features.

But even in the seemingly balanced experiments, there were differences. The in-domain results (italic) are obviously higher than the out-of-domain ones (normal type). The reason here is that the differences between the two 'domains' were not actively participating, but the systematic differences within each domain were. In this specific case, test items could be from programs seen

Table 3. Accuracy using all features, listed per train-test combination. Typesetting identifies the relation between training and test items: bold face type represents aided recognition, italic in-domain, normal out-of-domain and underline hindered

Training / Test	BelA	BelB	DutA	DutB
BelA-DutA	*0.612*	0.572	*0.615*	0.555
BelA-DutB	**0.793**	0.417	0.298	**0.729**
BelB-DutA	0.355	**0.750**	**0.772**	0.346
BelB-DutB	0.599	*0.658*	0.526	*0.648*

Table 4. Accuracies using specific feature types: averaged over train-test relation. The figures are based on a forced choice, with random selection if features cannot choose (results averaged over 10 runs with different random number seeds)

Feature type	Aided	In-domain	Out-of-domain	Hindered
All features	0.761	0.633	0.563	0.354
Char *n*-grams (lex)	0.726	0.629	0.544	0.379
Token unigrams (lex)	0.744	0.623	0.537	0.358
Token unigrams (abs)	0.514	0.504	0.503	0.497
Token bi/trigrams (lex)	0.752	0.622	0.557	0.358
Token bi/trigrams (abs)	0.629	0.541	0.529	0.427
Syntactic *n*-grams (lex)	0.750	0.617	0.552	0.345
Syntactic *n*-grams (abs)	0.585	0.528	0.520	0.456
Syntactic rewrites (abs)	0.577	0.524	0.516	0.461
All lex features	0.762	0.633	0.562	0.353
All abs features	0.639	0.549	0.536	0.426

during training, and program recognition became possible. As most shows had been subtitled completely in either Flanders or in The Netherlands, this then correlated with variety recognition. Training-test splits at the program level are also not sufficient, as similar programs may be mainly subtitled in one country, for example, the subtitles of the poker shows were split about 2:1 over Flanders and The Netherlands, and most cooking shows were subtitled in The Netherlands.

In Table 4, I show averaged results for each training-test relation, that is, the four measurements for each situation (bold, italic, normal, underlined in Table 3) are averaged. The top line (All features) corresponds to the results in Table 3. The averages give a better impression of the degree of domain influence. The in-domain result is comparable to the VarDial results, although obviously somewhat lower, probably due to the much smaller data set. If we relate the accuracies to a random choice, the out-of-domain result is only half as good as the in-domain one. Flemish and Netherlandic Dutch appear to be much less distinguishable than would seem on the basis of VarDial 2018 DFS. Still, some level of recognizability does exist. However, even this recognizability is exaggerated here by other properties of the Subtiel data, as we will see in the following.

The next eight lines of Table 4 show results when only using one specific type of features (listed in Table 2). On out-of-domain recognition, which is most related to actual language variety

Table 5. Most distinctive tokens in aided recognition, ordered by Contrib, that is, the contribution of the token in correct recognition (for calculation, see text). Counts are the absolute counts in the blocks BelA/BelB and DutA/DutB

Token	Gloss	Contrib	Counts
!	!	166,384	11/0 289/107
flop	flop	18,619	3/96 0/43
Komaan	Come on	14,236	72/67 1/0
Sami	Sami	14,091	84/10 0/0
STEM	VOICE	13,283	0/0 82/5
callt	calls	12,172	0/71 0/34
euro	euro	11,691	3/52 17/195
MUZIEK	MUSIC	11,487	0/0 66/39
gerecht	dish	10,373	4/39 5/211
callen	to call	10,163	0/64 0/33
aas	ace	9926	3/200 4/57
Frasier	Frasier	9746	70/1 14/0
all	all	9571	1/136 2/39
blinds	blinds	9039	0/66 0/16
Brooke	Brooke	8878	30/2 140/8

differences, the lexicalized token bi/trigrams appear the most powerful, closely followed by lexicalized syntactic bi/trigrams. On the in-domain tests, however, character *n*-grams are doing best, hardly worse than all features combined. They apparently best capture the (lexical) in-domain differences.

The last two lines in Table 4 show the importance of having access to lexical information, listing results for all lexicalized features combined and all abstract features combined. The lexicalized features alone reach the same quality as all features together, implying that most of the differences between Flemish and Netherlandic Dutch are centred around lexical items. Still, there are also differences at the abstract syntactic level. Furthermore, the abstract features are much less influenced by the domain. Domain influence too is mostly centred on lexical items. But here as well, there are differences, both between the two 'domains' and between programs within the domains. This should not come as a surprise, as we are dealing with various genres.

5. Qualitative analysis

In the previous section, I concluded that the major differences between both the language varieties and between the domains are centred on lexical items. For that reason, I will not investigate syntactic differences any further in this paper, but focus on token unigrams to exemplify the kinds of differences at play.

In Tables 5–7, I list the 15 features contributing most to the correct recognition of Flemish/Netherlandic items in the aided, in-domain and out-of-domain scenarios. To determine this contribution, a feature received a positive score corresponding to its odds every time it contributed to an item in line with its odds (e.g. a Netherlandic item if the odds favoured Netherlandic), but a negative score of three times its odds every time it contributed to an item not in line with its odds. A high position on the list is therefore a combination of high odds,

Table 6. Most distinctive tokens in in-domain recognition, ordered by Contrib, that is, the contribution of the token in correct recognition (for calculation, see text). Counts are the absolute counts in the blocks BelA/BelB and DutA/DutB

Token	Gloss	Contrib	Counts
!	!	29,060	11/0 289/107
Ridge	Ridge	22,277	1/6 153/14
Sami	Sami	14,091	84/10 0/0
Komaan	Come on	13,514	72/67 1/0
STEM	VOICE	13,283	0/0 82/5
MUZIEK	MUSIC	11,487	0/0 66/39
Oke	Right	5860	0/0 46/30
Forrester	Forrester	5935	0/0 55/2
DiMera	DiMera	5091	51/1 0/0
Stefano	Stefano	5073	51/1 0/5
Text	Text	4572	0/0 45/18
EEN	A	4220	0/0 34/32
Tuig	Gear	3723	0/44 0/0
Thorne	Thorne	3596	0/13 43/3
Xena	Xena	3595	3/0 0/43

frequency and precision on the test set. For each feature, the table shows the contribution as well as the absolute counts in the blocks BelA/BelB and DutA/DutB.[v]

Depending on the type of training and test material we are choosing, we arrive at quite different conclusions about the differences between Flemish and Netherlandic Dutch. With aided recognition, admittedly an unlikely setup but imaginable in circumstances of data sparseness, we observe mostly domain differences. The top-15 contains a whole set of poker terms: *flop*, *callt*, *callen*, *aas*, *all*[w] and *blinds*. Depending on the exact correspondence between domain and variety, these poker terms would be assigned to either Flemish or Netherlandic. The same is true for *euro*, *gerecht*, *Frasier* and *Brooke*. Whereas *euro* is spread out over many programs, but mostly occurs in domain B, *gerecht* is concentrated in cooking shows, all but one of the occurrences of *Frasier* are from Cheers or Frasier, and more than half the occurrences of *Brooke* from The Bold and the Beautiful. These explanations mean that only 5 of the top-15 win their position on the basis of variety differences alone.

With in-domain recognition, the differences between domains disappear, but are replaced by in-domain differences between programs, as witnessed by a number of names: *Ridge* (153× in The Bold and the Beautiful), *Sami* (84× in The Days of our Lives), *Forrester* (55× in The Bold and the Beautiful), *DiMera* (51× in Days of our Lives), *Stefano* (51× in Days of our Lives), *Thorne* (43× in The Bold and the Beautiful) and *Xena* (42× in Xena: Warrior Princess). The word *Tuig* (43× in Heavy Gear: the Animated Series) is similar, being a translation of the term 'Gear'. Together these cases form about half the top-15. The other half appear to genuine differences between the two language varieties, which I turn to now.

[v] A full list of feature counts and contributions can be found at https://cls.ru.nl/staff/hvhalteren/vardial_revisited/featcounts.txt and https://cls.ru.nl/staff/hvhalteren/vardial_revisited/contributions.txt.

[w] In all but six occurrences in domain B, *all* forms part of *all in*

Table 7. Most distinctive tokens in out-of-domain recognition, ordered by Contrib, that is, the contribution of the token in correct recognition (for calculation, see text). Counts are the absolute counts in the blocks BelA/BelB and DutA/DutB

Token	Gloss	Contrib	Counts
!	!	57,595	11/0 289/107
Komaan	Come on	14,069	72/67 1/0
MUZIEK	MUSIC	10,296	0/0 66/39
Oke	Right	5520	0/0 46/30
EEN	A	4351	0/0 34/32
Sami	Sami	3359	84/10 0/0
Text	Text	3240	0/0 45/18
enten	-separated-	1886	27/34 2/0
ine	-separated-	1791	14/32 0/0
STEM	VOICE	1639	0/0 82/5
ele	-separated-	1427	21/32 0/2
LACHT	LAUGHS	1367	0/0 18/19
Da's	That is	1354	109/100 11/11
ZE	SHE	1248	0/0 26/12
GELACH	LAUGHTER	971	0/0 27/9

With out-of-domain recognition, I finally hoped to see differences between Flemish and Netherlandic Dutch. But here too we first have to set apart a number of differences that are rather caused by differences in policy and/or habits between the two branches of BTI. The six upper case words in the top-15 stem from commentary on background noises, most likely because the subtitles are aimed at the hard of hearing. In the items selected for the current experiments, we see this type of text exclusively in subtitles from the Dutch branch. Related to this might be the exceptional use of the exclamation mark on the Netherlandic side: this might be a technique to show intonation more explicitly in the subtitles. Also sound related are hesitation markers such as *eh*, *uh* and *um*, which occur almost exclusively on the Netherlandic side. Another type of special features is formed by *enten*, *ine* and *ele*, which are all word endings starting with diacritics for words which I failed to correct during normalization, for example, *pati enten*, *hero ine* and *financi ele*. That they are found predominantly on the Flemish side implies that the Flemish branch of BTI uses diacritics far more than the Dutch branch. Finally, *Text* (and further in the list also *Broadcast* and *International*) stems from (sometimes partially) unremoved statements that the translation was made by *Broadcast Text International*, always on the Netherlandic side of the data. I do not expect any of these branch differences to be representative of any genuine differences between Flemish and Netherlandic Dutch.

The same is true for coincidences in the selection of programs subtitled by the two branches. A prime example is *Sami*. In our VarDial paper (van Halteren and Oostdijk 2018), we explained its high position by observing that Sami is a character in the soap Days of Our Lives, which happened to be translated in Flanders. In our current counts, we observe that 10 of the 94 occurrences are in domain B, so not in Days of Our Lives. It turns out that the name Sami also occurs in the shows How Did You Get Here and Survival of the Richest, but these too happen to be translated in Flanders. The same dismissal can be applied to most names, although one might argue for

Table 8. Hypothesized alternations. Ordered by rank in the distinctiveness list for out-of-domain recognition. Counts are the absolute counts in the blocks BelA/BelB and DutA/DutB

Token	Rank	Counts	Gloss	Alternative	Counts
Oke	4	0/0 46/30	Right	Goed	1183/839 798/654
percent	19	23/25 4/0	percent	procent	55/58 78/98
plots	21	43/42 6/3	suddenly	ineens	59/63 122/82
Zwijg	22	31/15 5/0	Be silent!	Stil	32/26 47/30
amuseren	36	35/13 5/1	amuse	vermaken	10/8 21/9
sofa	39	21/13 2/1	sofa	bank	83/67 137/105
job	49	27/22 3/3	job	baan	181/121 251/110
gsm	75	30/50 5/11	mobile phone	mobiel	15/11 17/27
nadien	76	6/7 0/0	afterwards	daarna	87/88 105/131
parking	85	6/6 0/0	parking	parkeerterrein	1/5 2/7
misgaan	89	1/2 7/20	fail	falen	7/7 2/4
weldra	96	4/8 0/0	soon	binnenkort	24/18 34/35

genuine differences in cases of clearly Flemish and Netherlandic names such as *Brussel* (3/9 0/2) and *Amsterdam* (0/0 9/11). And I also suspect coincidence for some other words: *client* ('client'; 3/43 0/1), *maximaal* ('at most'; 0/1 3/21), *gezochte* ('sought for'; 0/3 2/21) and *Waanzinnig* ('Crazy'; 0/2 5/20).

In order to find more (potentially) genuine differences than the remaining *Komaan*, *Oke* and *Da's*, we have to look beyond the top-15 of out-of-domain distinctive features. Then we are able to make some more interesting observations. Some of these are individual. *Da's* (rank 13) is an enclitic representation of the reduced form of *Dat is* ('That is') and hence represents spoken language. It occurs mainly on the Flemish side of the data (rank 30; 109/100 11/11), as does the lower case version (30/35 4/3). Unexpectedly, the full version *Dat is* also shows a bias towards the Flemish side (1934/1856 1893/1733). It appears that the Flemish subtitlers not only show the reduction more, but they even use this sentence start more often. Another observation relates to morphology. Dutch has two main plural noun suffixes: *-en* and *-s*. Most words only use one of these. In our data, we now observe that the plural of *leraar*, normally *leraren* (12/6 9/7), can in Flanders also be *leraars* (rank 107; 6/5 0/0). It may be that this is due to a single subtitler, but the 11 cases are spread over 9 programs covering various subdomains, which casts doubt on this explanation. Finally, we notice *halfuur* ('half hour'; rank 59; 26/32 4/5) and *uiteten* ('eat out'; rank 79; 14/7 2/0), which, in standard Dutch, both would be written as two words: *half uur* (10/20 22/59) and *uit eten* (33/12 57/19).

Other observations can be grouped; for example, Table 8 shows a number of word pairs which on the basis of our data might considered to be alternations, all taken from the top-100 ranks of the distinctiveness table for out-of-domain recognition. The word on the left was found as a side effect of recognition; the word on the right by considering intuitively likely alternatives. Apart from *percent* versus *procent*, none of these cases is straightforward. There may be ambiguity, as in *bank*, which also means 'bank'. Or the alternative depends on the context, as in *job*, where the alternative could also be *klus* when it concerns a short irregular job (19/30 33/48). Another complication is that single words can be in alternation with *n*-grams. An example here is *Zwijg*, which can also be expressed as *Hou je mond* (lit. 'hold your mouth'; 40/43 29/13).

Table 9. Probable explanations for tokens being in the top-100 of the distinctiveness list for out-of-domain recognition

Explanation	Number	Examples
BTI: Sounds	17	!, MUZIEK, eh
BTI: Orthography	9	ele, enten, 000
BTI: Non-subtitle	4	Text, Broadcast, Vertaling
Data selection: Names	32	Sami, Californie, Raul
Data selection: Other	7	client, gnocchi, moordenares
Culture: National names	2	Amsterdam, Utrecht
Variety: morphology	5	da's, leraars, halfuur
Variety: alternations	16	Komaan, percent, plots
Variety: other	8	kilogram, euro, Hee

Alternations between more than two options complicate things even more in some other cases in the top-100. *Komaan* at rank 2 (72/67 1/0) is (originally) a kind of encouraging opening of a suggestion for action ('let's'). In this sense, alternatives are *Kom op* (194/189 225/230), *Allee* (3/23 0/0) and *Vooruit* (118/130 92/79). In this group, *Komaan* and *Allee* are clearly marked as Flemish, whereas the others merely show trends. Another case are the options for 'or', namely *of*, *ofwel* and *oftewel* (and their capitalized versions), which have slight nuances in meaning but can generally replace each other. We consider this alternation because *Ofwel* is found at rank 37 in the list. In the lower case version, *of* is most frequent (2113/2239 2198/2470), as expected, and shows a slight bias towards Netherlandic Dutch. Both *ofwel* (9/12 4/4) and *oftewel* (5/6 3/2) lean towards Flemish Dutch. To the ears of the author (being Dutch), these forms sound slightly archaic, which is in line with a similar impression for *plots*, *nadien* and *weldra* in Table 8, potentially indicating a preference for more traditional forms in Flemish Dutch. However, the capitalized versions *Of* (423/375 483/383), *Ofwel* (10/18 0/2) and *Oftewel* (1/0 6/7) immediately cast doubt on such overgeneralization, as the most archaic form *Oftewel* points towards Netherlandic Dutch. A final larger alternation is found at rank 78: *Excuseer* (63/49 13/12), an apology at the start of a sentence. Obvious alternatives here are the popular English loanword *Sorry* (678/404 629/326) and the French loanword *Pardon* (86/45 84/65). Apart from *Pardon* in domain B, all lean towards Flemish Dutch. Possibly the Flemish are just more polite than the Dutch, or at least subtitle more polite programs.

Not all distinctive words are easily explained, however. After the deliberations above, we are still left with eight words: *kilogram* ('kilogram'; rank 41; 11/15 0/2), *massa's* ('masses'; rank 46; 9/12 1/0), *euro* ('euro'; rank 48; 3/52 15/195), *prikt* ('sticks'; rank 64; 0/0 7/7), *voorbije* ('past'; rank 65; 12/11 1/1), *impact* ('impact'; rank 66; 8/6 0/0), *Wow* ('Wow'; rank 83; 17/13 3/1) and *Hee* ('Hey'; rank 93; 6/1 31/8). An explanation for these might be found through careful analysis of their contexts and a more systematic search for alternatives. However, I cannot rule out, especially for the cases with very low counts, that we are simply dealing with author-based preferences. Unfortunately, metadata about the identity of the author is lacking, so that I cannot test this hypothesis.

The number of tokens for which the presented explanations are most probably valid are listed in Table 9. With 39 out of the 100 most distinguishing tokens indicating genuine differences between the two language varieties, I have to conclude that the accuracy of 0.563 for out-of-domain recognition still overestimates the differences. As to leads to find the differences, however, things are not as bleak as they may seem; 39 is not that low, and we can expect the artefacts to be found mostly

at the top of the list, so the range below that should provide a higher concentration of genuine differences.

6. Discussion

In this paper, I investigated to what degree language variety recognition and the identification of distinguishing features are helped or hindered by domain influences. Using a subset of the data used for the VarDial 2018 DFS task, extended with metadata, I performed experiments in which the relation between training and test data was controlled.

I found that domain can have an enormous influence on the results. Using training data from different domains for our two language varieties, Flemish and Netherlandic Dutch, leads to either exaggerated recognition quality (when test data are chosen in the same configuration) or behaviour significantly worse than random (when test data are chosen in the reversed configuration). Examining distinguishing features in the exaggerated scenario yields mostly misleading information in the sense that most identified features are domain dependent rather than variety dependent.

Even when using training material taken from the same general domain, test material from that same domain yields an overestimation of the variety differences. The reason is that now the various sources within the domain, in our case movies and TV shows, are being recognized. This effect is strongest with proper names, but for sure also extends beyond. The DFS task at VarDial 2018 used training and test material in such a configuration and I have to conclude that the winning 66% F1-score leads to a serious overestimation of the differences between Flemish and Netherlandic Dutch.[x] It would seem that the effects of bias are not weak enough to be 'tolerable' as suggested by Malmasi *et al.* (2015) on the basis of experiments on their data, but that serious caution has to be advised as was done in authorship recognition circles (Mikros and Argiri 2007).

A much more realistic estimate is provided when test data are taken from a different domain than the training data, even though this may lead to underestimation because in-domain differences are not taken into account. In my experiments in this paper, the recognition quality improvement over random choice with out-of-domain tests was only half that obtained with in-domain tests. And even this estimate may be too high, as there were quite a few distinguishing features that were unrelated to genuine language variety differences.

Interestingly, which feature type was most effective depended on the experimental setup. Whereas in-domain testing identified character *n*-grams as the best performing feature type, out-of-domain testing gave this qualification to (lexicalized) token bi/trigrams and syntactic *n*-grams. As the best feature type is a popular result to report, and differences in opinion here are not unknown, the respective studies could profit from investigating any biases that might be affecting the outcome.

Although even the out-of-domain tests led to a not quite realistic impression of the differences between Flemish and Netherlandic Dutch, these tests did provide feature information that can support a qualitative analysis of the differences. In Section 5, I discussed a number of lexical differences, and I assume that some syntactic differences might well be unearthed in a similar manner.

The experiments in this paper again demonstrated that one should always closely examine the data in any give classification task and identify (and try to undo) any influences from other differences than the ones sought for. Only then will it be possible to gain insight into the differences between the intended classes.

[x] It is unclear how the F1-score of 0.92 claimed by van der Lee and van den Bosch (2017) should be interpreted. As they do not describe clearly how they split their training and test data, it is hard to judge how much bias might be at work. However, their test items were not around 30 words, but rather subtitles for full movies or TV show episodes. It may well be that such a long document would include sufficient clues for such a high score.

References

Aguilar G., AlGhamdi F., Soto V., Solorio T., Diab M. and Hirschberg J. (2018). *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*. Melbourne, Australia: Association for Computational Linguistics.

Argamon S., Šarić M. and Stein S. S. (2003). Style mining of electronic messages for multiple author discrimination. In *Proceedings of ACM Conference on Knowledge Discovery and Data Mining, 2003*.

Baayen H., van Halteren H., Neijt A. and Tweedie F. (2002). An experiment in authorship attribution. In *Proceedings of JADT 2002: Sixth International Conference on Textual Data Statistical Analysis*, pp. 29–37.

Blondel V.D., Guillaume J.-L., Lambiotte R. and Lefebvre E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* **2008**(10), 10008.

Bouma G., Van Noord G. and Malouf R. (2001). Alpino: wide-coverage computational analysis of Dutch. *Language and Computers* 37, 45–59.

Campigotto R., Conde Céspedes P. and Guillaume J.-L. (2014). A generalized and adaptive method for community detection. arXiv preprint arXiv:1406.2518.

Chaski C.E. (2001). Empirical evaluations of language-based author identification techniques. *Forensic Linguistics* **8**(1), 1–65.

Çöltekin Ç., Rama T. and Blaschke V. (2018). Tübingen-Oslo team at the VarDial 2018 evaluation campaign: an analysis of n-gram features in language variety identification. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial), Santa Fe, USA*, pp. 55–65.

De Caluwe J. (2012). Dutch in Belgium facing multilingualism. In Hüning, M., Vogl, U., & Moliner, O. (Eds.). (2012). *Standard Languages and Multilingualism in European History*, (Vol. 1). John Benjamins Publishing, pp. 259–282.

Grondelaers S. and van Hout R. (2011). The standard language situation in the low countries: top-down and bottom-up variations on a diaglossic theme. *Journal of Germanic Linguistics* **23**(3), 199–243.

Hanani A., Qaroush A. and Taylor S. (2017). Identifying dialects with textual and acoustic cues. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects*, pp. 93–101.

Jauhiainen T., Lui M., Zampieri M., Baldwin T. and Lindén K. (2018). Automatic language identification in texts: a survey. arXiv preprint arXiv:1804.08186.

Komen E. (2015). Surfacing Dutch syntactic parses. Presentation at Computational Linguistics in the Netherlands (CLIN26), Amsterdam, 2015. http:wordpress.let.vupr.nlclin26abstracts.

Koppel M., Schler J. and Argamon S. (2009). Computational methods in authorship attribution. *Journal of the American Society for information Science and Technology* **60**(1), 9–26.

Malmasi S., Refaee E. and Dras M. (2015). Arabic dialect identification using a parallel multidialectal corpus. In *Conference of the Pacific Association for Computational Linguistics*, pp. 35–53.

Malmasi S., Zampieri M., Ljubešić N., Nakov P., Ali A. and Tiedemann J. (2016). Discriminating between similar languages and Arabic dialect identification: a report on the third DSL shared task. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pp. 1–14.

Mikros G.K. and Argiri E.K. (2007). Investigating topic influence in authorship attribution. In *Proceedings of the Workshop on Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection, SIGIR '07, Amsterdam*.

Molina G., AlGhamdi F., Ghoneim M., Hawwari A., Rey-Villamizar N., Diab M. and Solorio T. (2016). Overview for the second shared task on language identification in code-switched data. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pp. 40–49.

Murthy K.N. and Kumar B. (2006). Language identification from small text samples. *Journal of Quantitative Linguistics* **13**(1), 57–80.

Mustonen S. (1965). Multiple discriminant analysis in linguistic problems. *Statistical Methods in Linguistics* **4**, 37–44.

Solorio T., Blair E., Maharjan S., Bethard S., Diab M., Ghoneim M., Hawwari A., AlGhamdi F., Hirschberg J., Chang A. and Fung P. (2014) Overview for the first shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pp. 62–72.

Stamatatos E. (2009). A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology* **60**(3), 538–556.

van den Bosch A., Busser B. and Daelemans W. (2007). An efficient memory-based morphosyntactic tagger and parser for Dutch. In van Eynde F., Dirix P., Schuurman I. and Vandeghinste V. (eds), *Selected Papers of the 17th Computational Linguistics in the Netherlands Meeting, Leuven, Belgium*, pp. 99–114.

van der Lee C. (2017). *Text-Based Video Genre Classification Using Multiple Feature Categories and Categorization Methods*. Master's Thesis, Tilburg University.

van der Lee C. and van den Bosch A. (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), Valencia, Spain*, pp. 190–199.

van Halteren H. (2007). Author verification by linguistic profiling: an exploration of the parameter space. *ACM Transactions on Speech and Language Processing* **4**(1), 1–17.

van Halteren H. and Oostdijk N. (2018). Identification of differences between Dutch language varieties with the VarDial2018 Dutch-Flemish subtitle data. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial), Santa Fe, USA*, pp. 199–209.

van Halteren H. and Speerstra N. (2014). Gender recognition on Dutch tweets. *Computational Linguistics in the Netherlands Journal* **4**, 171–190.

Zampieri M., Tan L., Ljubešić N. and Tiedemann J. (2014). A report on the DSL shared task 2014. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, pp. 58–67.

Zampieri M., Tan L., Ljubešić N., Tiedemann J. and Nakov P. (2015). Overview of the DSL shared task 2015. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects*, pp. 1–9.

Zampieri M., Malmasi S., Ljubešić N., Nakov P., Ali A., Tiedemann J., Scherrer Y. and Aepli N. (2017). Findings of the VarDial evaluation campaign 2017. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pp. 1–15.

Zampieri M., Malmasi S., Nakov P., Ali A., Shon S., Glass J., Scherrer Y., Samardžić T., Ljubešić N., Tiedemann J., van der Lee C., Grondelaers S., Oostdijk N., van den Bosch A., Kumar R., Lahiri B. and Jain M. (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*, pp. 1–17.

Cite this article: van Halteren H. Domain bias in distinguishing Flemish and Dutch subtitles. *Natural Language Engineering* https://doi.org/10.1017/S1351324919000445