HeLI-based Experiments in Discriminating Between Dutch and Flemish Subtitles

Tommi Jauhiainen
University of Helsinki
@helsinki.fi

Heidi Jauhiainen
University of Helsinki
@helsinki.fi

Krister Lindén
University of Helsinki
@helsinki.fi

Abstract

This paper presents the experiments and results obtained by the SUKI team in the *Discriminating between Dutch and Flemish in Subtitles* shared task of the VarDial 2018 Evaluation Campaign. Our best submission was ranked 8th, obtaining macro F1-score of 0.61. Our best results were produced by a language identifier implementing the HeLI method without any modifications. We describe, in addition to the best method we used, some of the experiments we did with unsupervised clustering.

1 Introduction

The four first VarDial workshops have hosted several shared tasks concentrating on language identification of close languages or language varieties. The fifth VarDial workshop (Zampieri et al., 2018) introduced a new shared task concentrating on finding differences between the subtitles written in Netherlandic Dutch and Flemish Dutch (DFS). Netherlandic Dutch and Flemish Dutch are considered the same language by the ISO-639-3 standard since the Belgian dialect (Flemish) is only slightly different from the Dutch used in the Netherlands (Lewis et al., 2013). We had never experimented with the language identification of Dutch varieties and we were interested to see how well it can be done with the methods we have used in the past.

For the past five years we have been developing a language identifying method, which we call HeLI, for the Finno-Ugric Languages and the Internet project (Jauhiainen et al., 2015a). The HeLI method is a general purpose language identification method relying on observations of word and character *n*-gram frequencies from a language labeled corpus. The method is similar to Naive Bayes when using only relative frequencies of words as probabilities. Unlike Naive Bayes, it uses a back-off scheme to calculate the probabilities of individual words if the words themselves are not found in the language models. The optimal combination of language models used with the back-off scheme depend on the situation and is determined empirically using a development set. The choice is affected for example by the number and type of languages and the amount of training material. The back-off scheme begins from the most rarely seen features and backs off to more common features. We have participated in the shared tasks of three previous VarDial workshops (Zampieri et al., 2015; Malmasi et al., 2016; Zampieri et al., 2017) with language identifiers using the HeLI method or its variations (Jauhiainen et al., 2015b; Jauhiainen et al., 2016; Jauhiainen et al., 2017a). The method has turned out to be robust and competitive with other state-of-the-art language identification methods. For the current workshop, we wanted to try out some more variations and possible improvements to the original method. In addition to the adaptive language models we experimented with unsupervised clustering.

2 Related Work

Language identification is a task related to general text categorization and many of the methods are the same or similar to those used in that field. For more information on language identification and the

Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects, pages 137–144
Santa Fe, New Mexico, USA, August 20, 2018.

methods used for it, see the recent survey by Jauhiainen et al. (2018).

The language identification of Dutch, its varieties and the languages close to it has been considered earlier outside the VarDial context. Trieschnigg et al. (2012) evaluated rank- and cosine-similarity (nearest neighbour and nearest prototype) based language identification methods on the Dutch Folktale Database (Meder, 2010). The version of the database they used contained 15 languages or dialects close to Dutch, as well as English. The best macro F-score for document size identifications, 0.63, was obtained by the nearest neighbor cosine similarity method trained on word unigrams. Tulkens et al. (2016) used word2vec word embeddings in dialect identification of Dutch varieties.

Afrikaans is a close language to Dutch spoken in South Africa. The language identification between Afrikaans and Dutch has been examined several times in the past. Cowie et al. (1999) evaluated the common word, minimal distance, rank distance (Cavnar and Trenkle, 1994), and their own LZC methods and note that all of them were able to distinguish relatively well between the two languages. When automatically creating language trees from the universal declarations of human rights, Benedetto et al. (2002) group Afrikaans and Dutch together, with both equally related to Frisian. Singh (Singh, 2006; Singh, 2010) lists Dutch and Afrikaans as confusable languages with each other and Lui (2014) noticed that Afrikaans was confused especially with West Frisian. The latest study on Dutch language identification by van der Lee and van den Bosch (2017) led to the current shared task.

2.1 Unsupervised clustering

Unsupervised clustering of text aims to form coherent groups by gathering similar texts together. One of the first unsupervised clustering approaches to language identification task was presented by Biemann and Teresniak (2005). They use the co-occurrences of words to group words together in order to form a vocabulary of a language. Their method was later evaluated by Shiells and Pham (2010).

3 Task setup and data

To prepare for the shared task, the participants were provided with training and development datasets. The training set consisted of 150,000 lines for each of the Dutch varieties. In the other shared tasks of the VarDial workshops, the datasets have mostly contained only one sentence per line. The dataset for Dutch varieties, however, usually contained several sentences per line. The development part was quite small compared with the training data, only 250 lines per variety. However, the test set was comparably large: 10,000 lines for each language. This subtitles dataset and the methods used for collecting it are described in detail by van der Lee and van den Bosch (2017).

Participants were allowed to submit three runs for the DFS task. We submitted two, one with the original HeLI method, and one using HeLI with language model adaptation.

4 HeLI method in Discriminating between Dutch and Flemish, run 1

The HeLI method was first presented by Jauhiainen (2010) and later more formally by Jauhiainen et al. (2016). The description presented below differs from the original mostly in that we are leaving out the cut-off value c for the size of the language models. In this years shared tasks we found, and have already noticed it earlier, that if the corpus used as the training material is of good quality it is generally advisable to use all the available material. Furthermore, the penalty value compensates for some of the possible impurities in the language models. Leaving out the cut-off value negates the need for using the derived corpus C' in the equations. We also use both the original and lowercased versions of the words and n-grams as different language models, as using the original words was clearly beneficial with the development set. We present the complete formulas here as used by the best submitted run in order to make this article as self contained as possible.

Description of the used version of the HeLI method The goal is to correctly guess the language $g \in G$ for each of the lines in the test set. In the HeLI method, each language g is represented by several different language models only one of which is used for every word t in the line M. The language models are: a model based on words and one or more models based on character n-grams from one to n_{max}. For the DFS task, we used n_{max} up to eight. When we encounter a word not found in the word-based language models, we back off to using the *n*-grams of the size n_{max}. If we are unable to apply the *n*-grams of the size n_{max}, we back off to lower order *n*-grams and, if needed, we continue backing off until character unigrams. As both original and lowercased models are used, the models with original words are used first. If the back-off function is needed, we back of to lowercased words, then to original *n*-grams and then to lowercased *n*-grams. When backing from original *n*-grams to lowercased *n*-grams, the current implementation first backs off all the way to unigrams of original characters before moving on to lowercased *n*-grams of the size n_{max}, practically dropping the lowercased *n*-grams out of the equation.

The training data is tokenized into words using non-alphabetic and non-ideographic characters as delimiters. If lowercased language models are being created, the data is lowercased. The relative frequencies of the words are calculated. Also the relative frequencies of character *n*-grams from 1 to n_{max} are calculated inside the words, so that the preceding and the following space-characters are included. The *n*-grams are overlapping, so that for example a word with three characters includes three character trigrams. Word *n*-grams were not used, so all subsequent references to *n*-grams in this article refer to *n*-grams of characters. Then we transform those relative frequencies into scores using 10-based logarithms. Among the language models generated from the DFS corpus, the largest model was for original (non-lowercased) character 7-grams, including 333,256 different 7-grams for Dutch.

$dom(O(C))$ is the set of all words found in the models of any language $g \in G$. For each word $t \in dom(O(C))$, the values $v_{C_g}(t)$ for each language g are calculated, as in Equation 1.

$$v_{C_g}(t) = \begin{cases} -\log_{10}\left(\frac{c(C_g,t)}{l_{C_g}}\right) & \text{, if } c(C_g,t) > 0 \\ p & \text{, if } c(C_g,t) = 0, \end{cases} \tag{1}$$

where $c(C_g, t)$ is the number of words t and l_{C_g} is the total number of all words in language g. If $c(C_g, t)$ is zero, then $v_{C_g}(t)$ gets the penalty value p. The penalty value has a smoothing effect in that it transfers some of the probability mass to unseen features in the language models.

The corpus containing only the *n*-grams in the language models is called C^n. The domain $dom(O(C^n))$ is the set of all character *n*-grams of length n found in the models of any language $g \in G$. The values $v_{C^n_g}(u)$ are calculated similarly for all *n*-grams $u \in dom(O(C^n))$ for each language g, as shown in Equation 2.

$$v_{C^n_g}(u) = \begin{cases} -\log_{10}\left(\frac{c(C^n_g,u)}{l_{C^n_g}}\right) & \text{, if } c(C^n_g,u) > 0 \\ p & \text{, if } c(C^n_g,u) = 0, \end{cases} \tag{2}$$

where $c(C^n_g, u)$ is the number of *n*-grams u found in the corpus of the language g and $l_{C^n_g}$ is the total number of the *n*-grams of length n in the derived corpus of language g. These values are used when scoring the words while identifying the language of a text.

When using *n*-grams, the word t is split into overlapping *n*-grams of characters u^n_i, where $i = 1, ..., l_t - n$, of the length n. Each of the *n*-grams u^n_i is then scored separately for each language g in the same way as the words.

If the *n*-gram u^n_i is found in $dom(O(C^n_g))$, the values in the models are used. If the *n*-gram u^n_i is not found in any of the models, it is simply discarded. We define the function $d_g(t, n)$ for counting *n*-grams in t found in a model in Equation 3.

$$d_g(t,n) = \sum_{i=1}^{l_t-n} \begin{cases} 1 & \text{, if } u^n_i \in dom(O(C^n)) \\ 0 & \text{, otherwise} \end{cases} \tag{3}$$

When all the *n*-grams of the size n in the word t have been processed, the word gets the value of the average of the scored *n*-grams u^n_i for each language, as in Equation 4.

$$v_g(t,n) = \begin{cases} \frac{1}{d_g(t,n)} \sum_{i=1}^{l_t-n} v_{C_g^n}(u_i^n) & \text{, if } d_g(t,n) > 0 \\ v_g(t,n-1) & \text{, otherwise,} \end{cases} \tag{4}$$

where $d_g(t,n)$ is the number of *n*-grams u_i^n found in the domain $dom(O(C_g^n))$. If all of the *n*-grams of the size n were discarded, $d_g(t,n) = 0$, the language identifier backs off to using *n*-grams of the size $n-1$. If no values are found even for unigrams, a word gets the penalty value p for every language, as in Equation 5.

$$v_g(t,0) = p \tag{5}$$

The mystery text is tokenized into words using the non-alphabetic and non-ideographic characters as delimiters. The words are lowercased when lowercased models are being used. After this, a score $v_g(t)$ is calculated for each word t in the mystery text for each language g. If the word t is found in the set of words $dom(O(C_g))$, the corresponding value $v_{C_g}(t)$ for each language g is assigned as the score $v_g(t)$, as shown in Equation 6.

$$v_g(t) = \begin{cases} v_{C_g}(t) & \text{, if } t \in dom(O(C_g)) \\ v_g(t, min(n_{max}, l_t+2)) & \text{, if } t \notin dom(O(C_g)) \end{cases} \tag{6}$$

If a word t is not found in the set of words $dom(O(C_g))$ and the length of the word l_t is at least $n_{max}-2$, the language identifier backs off to using character *n*-grams of the length n_{max}. In case the word t is shorter than $n_{max}-2$ characters, $n = l_t + 2$.

The whole line M gets the score $R_g(M)$ equal to the average of the scores of the words $v_g(t)$ for each language g, as in Equation 7 .

$$R_g(M) = \frac{\sum_{i=1}^{l_{T(M)}} v_g(t_i)}{l_{T(M)}} \tag{7}$$

where $T(M)$ is the sequence of words and $l_{T(M)}$ is the number of words in the line M. Since we are using negative logarithms of probabilities, the language having the lowest score is returned as the language with the maximum probability for the mystery text.

Results of the run 1 on the development and the test sets The development set was used for finding the best values for the parameters n_{max} and p. The recall-values for different combinations can be seen in Table 1.

Leaving out any of the models did not seem to move recall into a better direction from the 64.6% obtained when using all the available language models. We decided to use all the generated models with the penalty value of 7.7 for the first run. We included the development set in the training material to generate the final language models. The run on the test set reached the recall of 61.4%. The results on the test set are naturally somewhat worse than on the development set, as the parameters have not been optimized for it. The macro F1-score obtained was 0.61. The recall was clearly better for Flemish than for Dutch as can be seen in Table 2. The length of the lines to be identified ranged from 111 to 385 characters. The results indicate that the length of the sequence to be identified is not a major issue for the method as the average lengths were very similar for both correctly and incorrectly identified texts.

5 HeLI with adaptive language models, run 2

With adaptive language models, new information is introduced in the language models from unlabeled texts while they are being identified. Using adaptive language models with HeLI means that we first identify all the lines in the test corpus, then we determine which of our identifications is most probably correct. For guessing the correctness, we used the absolute difference between the scores of the two languages as given by the HeLI method. We then labeled the line with the largest difference as the winning

Original words	Original n_{max}	Lowercased words	Lowercased n_{max}	Penalty p	Recall
yes	8	yes	8	7.7	64.6%
yes	8	no	8	7.7	64.6%
yes	7	no	8	8.6	64.6%
yes	7	no	7	8.6	64.6%
yes	7	no	6	8.6	64.6%
yes	7	no	-	8.6	64.6%
yes	7	yes	8	8.5-8.6	64.4%
yes	8	yes	7	7.7	64.4%
yes	6	no	7	7.7-7.8/8.0-8.6	64.2%
yes	6	no	-	8.4	64.2%
yes	6	yes	8	7.7-7.8/8.0-8.6	64.0%
yes	7	no	-	8.5	64.0%
no	7	no	-	8.9	63.8%
no	8	yes	8	8.0	63.6%
no	-	yes	8	8.0	63.4%
no	6	no	-	8.4	63.0%
no	-	no	8	7.8-8.3/8.5-8.7/9.2-9.4	62.2%
no	5	no	-	8.9	61.8%
no	4	no	-	8.4	57.8%
no	-	no	4	7.9	55.0%

Table 1: Baseline HeLI recall in development data with different combinations of parameters.

Correct language	Identified language	Number	Average length in words	Average length in characters
DUT	DUT	5679	34	187
DUT	BEL	4321	34	178
BEL	BEL	6592	33	178
BEL	DUT	3408	34	185

Table 2: Baseline HeLI statistics for run 1.

language and added the words and character *n*-grams from that line to the corresponding language model. Then we used the adapted models to re-identify the remaining unlabeled lines and continued labeling one line at a time until all the lines were labeled. This method worked very well with German dialects and Indo-Aryan languages for the other two tasks we participated in the VarDial workshop. However, as can be seen in Table 3, using the adaptive language models did not change the results very much for the Dutch varieties.

Orig. words	Orig. n_{max}	Low. words	Low. n_{max}	Penalty p	Recall (dynamic)	Recall (original)
yes	8	yes	8	7.7	64.8%	64.6%
no	8	yes	8	8.0	63.6%	63.6%
no	-	no	4	7.9	54.8%	55.0%
no	-	no	8	7.8	61.6%	62.2%

Table 3: Recall in development with different combinations of language models.

We submitted the second, and our final run, to the DFS task using HeLI with adaptive language models and with the same parameters as for the first run. The recall on the test set was 61.15%, which did not improve on the first run. Similarly, the resulting F1-score of 0.6107 did not improve the score gained with the unmodified HeLI method.

6 Experiments with unsupervised clustering

We wanted to try out an idea that using unsupervised clustering on the test set before actual language identification might be beneficial. The idea is that the lines of the test set written in the same language might be more alike with each other than with the material used for training the language identifier. Grouping similar lines together would make it easier for the language identifier to identify the text as the length of the text to be identified is usually directly related to the accuracy of the identification (Jauhiainen et al., 2017b). To our knowledge, this strategy has not been used previously.

We decided to try clustering with an ad-hoc nearest neighbor clustering-method using the HeLI method as the similarity measure. In our nearest neighbor clustering each line is considered a separate language and language models are created for them. Each line is then scored using these models with the language model of the line itself omitted from the repertoire. After scoring, each line is grouped in the same group with the line whose model gave the best score. In this way each identified group would include at least two lines. We created a separate language model for each of the 500 lines in the development set. We tested clustering with lowercased character *n*-gram models. The results of the accuracy of any line being paired with a line from the same language can be seen in Table 4.

Lowercased n_{max}	**Penalty** p	**Accuracy**
1	5	55.8%
2	3	53.2%
3	6	59.0%
4	11	56.6%
5	11	57.0%
6	4	54.4%
7	4	55.0%
8	4	56.2%

Table 4: Clustering accuracy with different language models.

Character trigrams made the best clusters with accuracy of 59%. However, the grouping created by the unsupervised method seemed to be too random, so we concluded that identifying the groups would only make results worse and did not continue with these experiments.

7 Conclusions and future work

The non-discriminative nature of the HeLI method leaves it at a disadvantage against some of the more discriminative classification methods when the languages or dialects to be distinguished are extremely close. We did not use any discriminative features with the method for this shared task. In the future, we will continue experimenting with adding some discriminative elements to the HeLI method when dealing with very close languages.

Our adaptive language models fared very well in the two other tasks of this years evaluation campaign. We believe that the reasons the adaptive language models did not succeed so well in the DFS task are that the training corpus was already quite large and the test set was not from any one homogenous domain. If the mystery text would have been for example a set of subtitles for a single television series or a new movie, the adaptive language models could have learned the names used in the set from the more distinguishable lines and the names might have in turn helped with the more difficult lines.

The unsupervised clustering should be trialed on other datasets, and it might be applicable in an out-of-domain situation. We used only lowercased character *n*-grams for clustering and the effect of also using words should be verified. We, furthermore, experimented with only one unsupervised clustering method; it may not have been the best one and others should be evaluated.

Acknowledgments

This research was partly conducted with funding from the Kone Foundation Language Programme (Kone Foundation, 2012).

References

Dario Benedetto, Emanuele Caglioti, and Vittorio Loreto. 2002. Language Trees and Zipping. *Physical Review Letters*, 88(4).

Chris Biemann and Sven Teresniak. 2005. Disentangling from Babylonian confusion — Unsupervised Language Identification. In Alexander Gelbukh, editor, *Proceedings of the 6th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2005)*, pages 773–784, Mexico City, Mexico. Springer.

William B. Cavnar and John M. Trenkle. 1994. N-Gram-Based Text Categorization. In *Proceedings of SDAIR-94, Third Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, USA.

Jim Cowie, Yevgeny Ludovik, and Ron Zacharski. 1999. Language Recognition for Mono- and Multi-lingual Documents. In *Proceedings of the VexTal Conference*, pages 209–214, Venice, Italy.

Heidi Jauhiainen, Tommi Jauhiainen, and Krister Lindén. 2015a. The Finno-Ugric Languages and The Internet Project. *Septentrio Conference Series*, 0(2):87–98.

Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. 2015b. Discriminating Similar Languages with Token-Based Backoff. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (LT4VarDial)*, pages 44–51, Hissar, Bulgaria.

Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. 2016. HeLI, a Word-Based Backoff Method for Language Identification. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 153–162, Osaka, Japan.

Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. 2017a. Evaluating HeLI with Non-Linear Mappings. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 102–108, Valencia, Spain, April.

Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. 2017b. Evaluation of Language Identification Methods Using 285 Languages. In *Proceedings of the 21st Nordic Conference on Computational Linguistics (NoDaLiDa 2017)*, pages 183–191, Gothenburg, Sweden. Linköping University Electronic Press.

Tommi Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2018. Automatic Language Identification in Texts: A Survey. *arXiv preprint arXiv:1804.08186*.

Tommi Jauhiainen. 2010. Tekstin kielen automaattinen tunnistaminen. Master's thesis, University of Helsinki, Helsinki.

Kone Foundation. 2012. The language programme 2012-2016. http://www.koneensaatio.fi/en.

M. Paul Lewis, Gary F. Simons, and Charles D. Fennig, editors. 2013. *Ethnologue: Languages of the world, seventeenth edition*. SIL International, Dallas, Texas.

Marco Lui. 2014. *Generalized Language Identification*. Ph.D. thesis, The University of Melbourne.

Shervin Malmasi, Marcos Zampieri, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, and Jörg Tiedemann. 2016. Discriminating between Similar Languages and Arabic Dialect Identification: A Report on the Third DSL Shared Task. In *Proceedings of the 3rd Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (VarDial)*, Osaka, Japan.

Theo Meder. 2010. From a Dutch folktale database towards an international folktale database. *Fabula*, 51:6–22.

Karen Shiells and Peter Pham. 2010. Unsupervised Clustering for Language Identification. Project Report, Stanford University.

Anil Kumar Singh. 2006. Study of Some Distance Measures for Language and Encoding Identification. In *Proceedings of the Workshop on Linguistic Distances*, pages 63–72, Sydney, Australia.

Anil Kumar Singh. 2010. *Modeling and Application of Linguistic Similarity*. Ph.D. thesis, International Institute of Information Technology, Hyderabad.

Dolf Trieschnigg, Djoerd Hiemstra, Mariët Theune, Franciska de Jong, and Theo Meder. 2012. An Exploration of Language Identification Techniques for the Dutch Folktale Database. In *Proceedings of the LREC workshop Adaptation of Language Resources and Tools for Processing Cultural Heritage*, pages 47–51, Istanbul, Turkey.

Stéphan Tulkens, Chris Emmery, and Walter Daeleman. 2016. Evaluating Unsupervised Dutch Word Embeddings as a Linguistic Resource. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 4130–4136. European Language Resources Association (ELRA).

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, Valencia, Spain.

Marcos Zampieri, Liling Tan, Nikola Ljubešić, Jörg Tiedemann, and Preslav Nakov. 2015. Overview of the DSL Shared Task 2015. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (LT4VarDial)*, pages 1–9, Hissar, Bulgaria.

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)*, Valencia, Spain.

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.