Evaluating HeLI with Non-linear Mappings

Tommi Jauhiainen
University of Helsinki
@helsinki.fi

Krister Lindén
University of Helsinki
@helsinki.fi

Heidi Jauhiainen
University of Helsinki
@helsinki.fi

Abstract

In this paper we describe the non-linear mappings we used with the Helsinki language identification method, HeLI, in the 4^{th} edition of the Discriminating between Similar Languages (DSL) shared task, which was organized as part of the VarDial 2017 workshop. Our SUKI team participated in the closed track together with 10 other teams. Our system reached the 7^{th} position in the track. We describe the HeLI method and the non-linear mappings in mathematical notation. The HeLI method uses a probabilistic model with character *n*-grams and word-based back-off. We also describe our trials using the non-linear mappings instead of relative frequencies and we present statistics about the back-off function of the HeLI method.

1 Introduction

The 4^{th} edition of the Discriminating between Similar Languages (DSL) shared task (Zampieri et al., 2017) was divided into an open and a closed track. In the closed track the participants were allowed to use only the training data provided by the organizers, whereas in the open track the participants could use any data source they had at their disposal. This year we did not participate in the open track, so we did not use any additional sources for training and development. The creation of the earlier DSL corpora has been described by Tan et al. (2014). This year's training data consisted of 18,000 lines of text, excerpts of journalistic texts, for each of the 14 languages. The corresponding development set had 2,000 lines of text for each language. The task had a language selection comparable to the 1^{st} (Zampieri et al., 2014), 2^{nd} (Zampieri et al., 2015), and 3^{rd} (Malmasi et al., 2016) editions of the shared task. The languages and varieties are listed in Table 1. The differences from the previous year's shared task were the inclusion of Persian and Dari languages, as well as replacing the Mexican Spanish variety with Peruvian Spanish.

Country	Language
Bosnia and Herzegovina	Bosnian
Croatia	Croatian
Serbia	Serbian
Malaysia	Malay
Indonesia	Indonesian
Iran	Persian
Afghanistan	Dari
Canada	French
France	French
Brazil	Portuguese
Portugal	Portuguese
Argentina	Spanish
Spain	Spanish
Peru	Spanish

Table 1: The languages and varieties of the 4^{th} edition of the Discriminating between Similar Languages (DSL) shared task.

For the 4^{th} edition, we were interested in modifying the HeLI method and use the TF-IDF scores and some non-linear mappings instead of relative frequencies. We were inspired by the successful use of TF-IDF scores by Barbaresi (2016). He was able to significantly boost the accuracy of his identifier after the 3^{rd} edition of the shared task by using the TF-IDF scores. Earlier, Brown (2014) managed to boost several language identification methods using non-linear mappings.

2 Related Work

Automatic language identification of digital text has been researched for more than 50 years. The first article on the subject was written by Mustonen (1965), who used multiple discriminant anal-

Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects, pages 102–108,
Valencia, Spain, April 3, 2017. ©2017 Association for Computational Linguistics

ysis to distinguish between Finnish, English and Swedish. For more of the history of automatic language identification the reader is suggested to take a look at the literature review chapter of Marco Lui's doctoral thesis (Lui, 2014).

There has also been research directly involving the language groups present in this year's shared task. Automatic identification of South-Slavic languages has been researched by Ljubešic et al. (2007), Tiedemann and Ljubešic (2012), Ljubešic and Kranjcic (2014), and Ljubešic and Kranjcic (2015). Brown (2012) presented confusion matrices for the languages of the former Yugoslavia (including Bosnian and Croatian) as well as for Indo-Iranian languages (including Western and Eastern Farsi). Chew et al. (2009) experimented distinguishing between Dari and Farsi, as well as Malay and Indonesian, among others. Distinguishing between Malay and Indonesian was studied by Ranaivo-Malançon (2006). Automatic identification of French dialects was studied by Zampieri et al. (2012) and Zampieri (2013). Discriminating between Portuguese varieties was studied by Zampieri and Gebre (2012), whereas Zampieri et al. (2012), Zampieri (2013), Zampieri et al. (2013), and Maier and Gómez-Rodríguez (2014) researched language variety identification between Spanish dialects.

The system description articles provided for the previous shared tasks are all relevant and references to them are provided by Zampieri et al. (2014), Zampieri et al. (2015), and Malmasi et al. (2016). Detailed analysis of the first two shared tasks was done by Goutte et al. (2016).

The language identification method used by the system presented in this article, HeLI, was first introduced by Jauhiainen (2010) and it was also described in the proceedings of the 2^{nd} edition of the DSL shared task (Jauhiainen et al., 2015). The complete description of the method was first presented in the proceedings of the 3^{rd} VarDial workshop (Jauhiainen et al., 2016). The language identifier tool using the HeLI method is available as open source from GitHub[1]. The non-linear mappings evaluated in this article were previously tested with several language identifiers by Brown (2014).

3 Methodology

In this paper, we re-present most of the description of the HeLI method from the last year's system description paper (Jauhiainen et al., 2016). We leave out the mathematical description of the words as features, as they were not used in the submitted runs. We tried several combinations of words, lowercased words, *n*-grams, and lowercased *n*-grams with the development set. The best results of these trials can be seen in Table 2. In the table, ”l. n_{max}” refers to the maximum number of lowercased *n*-grams, ”c. n_{max}” to the *n*-grams with also capital letters, ”l. w.” to lowercased words, and ”c. w.” to words with original capitalization. We did similar tests with different combinations of the language models when choosing the models to be used with the loglike-function described later.

rec.	**l.** n_{max}	**c.** n_{max}	**l. w.**	**c. w.**
0.9107	0	8	no	no
0.9107	8	8	no	no
0.9099	0	8	yes	no
0.9098	8	0	yes	yes
0.9098	8	8	yes	yes
0.9092	0	8	no	yes
0.9060	8	8	yes	no
0.9059	8	0	yes	no
0.9052	8	0	no	no

Table 2: Testing the different combinations of language models on the development set.

3.1 On Notation

A corpus C is a finite sequence, $u_1, ..., u_l$, of individual tokens u_i, which may be words or characters. The total count of all individual tokens u in the corpus C is denoted by l_C. A feature f is some countable characteristic of the corpus C. When referring to all features F in a corpus C, we use C^F and the count of all features is denoted by l_{C^F}. The count of a feature f in the corpus C is referred to as $c(C, f)$. An *n*-gram is a feature which consists of a sequence of n individual tokens. An *n*-gram of the length n starting at position i in a corpus is denoted u_i^n. If $n = 1$, u is an individual token. When referring to all *n*-grams of length n in a corpus C, we use C^n and the count of all such *n*-grams is denoted by l_{C^n}. The count of an n-gram u in a corpus C is referred to as $c(C, u)$ and is defined by Equation 1.

$$c(C, u) = \sum_{i=1}^{l_C+1-n} \begin{cases} 1 & \text{, if } u = u_i^n \\ 0 & \text{, otherwise} \end{cases} \qquad (1)$$

[1] https://github.com/tosaja/HeLI

The set of languages is G, and l_G denotes the number of languages. A corpus C in language g is denoted by C_g. A language model O based on C_g is denoted by $O(C_g)$. The features given values by the model $O(C_g)$ are the domain $dom(O(C_g))$ of the model. In a language model, a value v for the feature f is denoted by $v_{C_g}(f)$. For each potential language g of a corpus C in an unknown language, a resulting score $R_g(C)$ is calculated. A corpus in an unknown language is also referred to as a mystery text.

3.2 HeLI Method

The goal is to correctly guess the language $g \in G$ in which the monolingual mystery text M has been written, when all languages in the set G are known to the language identifier. In the method, each language $g \in G$ is represented by several different language models based on character *n*-grams from one to n_{max}. Only one of the language models is used for every word t found in the mystery text M. The model used is selected by its applicability to the word t under scrutiny. If we are unable to apply the *n*-grams of the size n_{max}, we back off to lower order *n*-grams. We continue backing off until character unigrams, if needed.

A development set is used for finding the best values for the parameters of the method. The three parameters are the maximum length of the used character *n*-grams (n_{max}), the maximum number of features to be included in the language models (cut-off c), and the penalty value for those languages where the features being used are absent (penalty p). The penalty value has a smoothing effect in that it transfers some of the probability mass to unseen features in the language models.

3.2.1 Creating the Language Models

The training data is tokenized into words using non-alphabetic and non-ideographic characters as delimiters. 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.

The c most common *n*-grams of each length in the corpus of a language are included in the language models for that language. We estimate the probabilities using relative frequencies of the character *n*-grams in the language models, using only the relative frequencies of the retained tokens. Then we transform those frequencies into scores using 10-based logarithms.

The derived corpus containing only the *n*-grams retained 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 all languages $g \in G$. 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

$$v'_{C'^n_g}(u) = \begin{cases} -\log_{10}\left(v_{C_g}(u)\right) & \text{, if } c(C'^n_g, u) > 0 \\ p & \text{, if } c(C'^n_g, u) = 0 \end{cases} \qquad (2)$$

In the first run of the shared task we used relative frequencies of *n*-grams as values $v_{C_g}(u)$. They are calculated for each language g, as in Equation 3

$$v_{C_g}(u) = \frac{c(C'^n_g, u)}{l_{C'^n_g}} \qquad (3)$$

where $c(C'^n_g, u)$ is the number of *n*-grams u found in the derived 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.

Brown (2014) experimented with five language identifiers using two non-linear mappings, the gamma and the loglike functions. We tested applying the two non-linear mappings to the relative frequencies. Both functions have a variable (*gamma* or *tau*), the value of which has to be empirically found using the development set.

The value $v_{C_g}(u)$ using the gamma function is calculated as in Equation 4

$$v_{C_g}(u) = \left(\frac{c(C'^n_g, u)}{l_{C'^n_g}}\right)^{\gamma} \qquad (4)$$

The value $v_{C_g}(u)$ using the loglike function is calculated as in Equation 5

$$v_{C_g}(u) = \frac{\log(1 + 10^{\tau}\frac{c(C'^n_g, u)}{l_{C'^n_g}})}{\log(1 + 10^{\tau})} \qquad (5)$$

3.2.2 Scoring *N*-grams in the Mystery Text

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

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 6.

$$d_g(t, n) = \sum_{i=1}^{l_t+1-n} \begin{cases} 1 & \text{, if } u_i^n \in dom(O(C'^n)) \\ 0 & \text{, otherwise} \end{cases} \tag{6}$$

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_i^n for each language, as in Equation 7

$$v_g(t, n) = \begin{cases} \frac{1}{d_g(t,n)} \sum_{i=1}^{l_t+1-n} v'_{C'^n_g}(u_i^n) & \text{, if } d_g(t, n) > 0 \\ v_g(t, n-1) & \text{, otherwise} \end{cases} \tag{7}$$

where $d_g(t, n)$ is the number of *n*-grams u_i^n found in the domain $dom(O(C'^n_g))$. 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 8.

$$v_g(t, 0) = p \tag{8}$$

3.2.3 Language Identification

The mystery text is tokenized into words using the non-alphabetic and non-ideographic characters as delimiters. After this, a score $v_g(t)$ is calculated for each word t in the mystery text for each language g. If the length of the word l_t is at least $n_{max} - 2$, the language identifier uses 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 mystery text 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 9

$$R_g(M) = \frac{\sum_{i=1}^{l_{T(M)}} v_g(t_i)}{l_{T(M)}} \tag{9}$$

where $T(M)$ is the sequence of words and $l_{T(M)}$ is the number of words in the mystery text 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.

4 Experiments

In order to find the best possible parameters (n_{max}, *c*, and *p*), we applied a simple form of the greedy algorithm using the development set. The best recall for the original HeLI method, 0.9105, was reached using $n_{max} = 8$, *c* = 170,000, and *p* of 6.6.

4.1 TF-IDF

We made a small experiment trying to adapt the HeLI method to use TF-IDF scores (product of term frequency and inverse document frequency). TF-IDF scores were successfully used to boost the performance of a Naive Bayes identifier by Barbaresi (2016). Also Malmasi et al. (2015) used character *n*-grams from one to four, which were weighted with TF-IDF. There are several variations of TF-IDF weighting scheme and Malmasi et al. (2015) do not specify whether they used the basic formula or not. We calculated the TF-IDF as in Equation 10

$$v_{C_g}(u) = c(C_g, u) log \frac{l_G}{df(C_G, u)} \tag{10}$$

where *df*() is defined as in Equation 11. Let l_G be the number of languages in a language segmented corpus C_G. We define the number of languages in which an *n*-gram u appears as the document frequency *df* of u as

$$df(C_G, u) = \sum_{g=1}^{l_G} \begin{cases} 1 & \text{, if } c(C_g, u) > 0 \\ 0 & \text{, otherwise} \end{cases} \tag{11}$$

We used the $v_{C_g}(u)$ values from Equation 10 instead of relative frequencies in Equation 2, but we were unable to come even close to the accuracy of our original method. We did not submit a run using the TF-IDF weighting.

4.2 Gamma Function

Using the gamma function in his experiments, Brown (2014) was able to reduce the error rate of his own language identifier by 83.9% with 1366 languages and 76.7% with 781 languages. We tested using the gamma function with the development set, which did not manage to improve our results. It seems that the penalty value p of the HeLI method and the γ variable have at least partly the same effect. If we fix one of the values we are able to reach almost or exactly the same results by varying the other. Table 3 shows some of the results on the development set. When using γ of 1.0 the method is identical to the original HeLI method. As there were no improvements on the results at all, we decided not to submit a run using the gamma function.

4.3 Loglike Function

Table 4 shows some of the results on the development set when using the loglike function, $n_{max} =$

Recall	Penalty p	Gamma γ
0.9105	*3.3*	*0.5*
0.9102	4.6	0.7
0.9103	5.3	0.8
0.9105	**6.6**	**1.0**
0.9104	7.9	1.2
0.9104	8.6	1.3
0.9105	*9.9*	*1.5*
0.9104	11.2	1.7

Table 3: Testing the gamma on the development set.

8, and c = 170,000. There seemed to be a local optimum at around $\tau = 2.9$, so we experimented with a bit different n_{max} and c around it as well. The best recall of 0.9109 was provided by $n_{max} = 7$, c = 180,000, and $\tau = 3.0$. The loglike funtion seemed to make a tiny (about half a percent) improvement on the error rate when using the development set. Using the loglike function, Brown (2014) was able to reduce the errors made by his own identifier by 83.8% with 1366 languages and 76.7% with 781 languages. Even though our error reduction was far from Brown's numbers, we still decided to submit a second run using the loglike function.

Recall	Penalty p	Tau τ
0.9104	6.5	0
0.9103	5.2	2.0
0.9104	4.7	2.7
0.9107	4.6	2.8
0.9106	4.5	2.9
0.9107	4.4	3.0
0.9104	4.3	3.2
0.9101	4.1	3.5
0.9075	3.0	4.5
0.9058	1.2	6.5

Table 4: Testing the loglike function on the development set.

5 Results

Our SUKI team submitted two runs for the closed track. For both of the runs we used all of the training and the development data to create the language models. The first run was submitted using the relative frequencies as in Equation 3. In the second run, we used the loglike function as in Equation 5. The results and the parameters for each run can be seen in Tables 5 and 6. We have also included the results and the name of the winning team CECL (Bestgen, 2017).

Run	Accuracy	F1 (macro)
CECL run1	0.9274	0.9271
SUKI run 2	0.9099	0.9097
SUKI run 1	0.9054	0.9051

Table 5: Results for the closed training.

Run	n_{max}	*c*	*p*
SUKI run 1	8	170,000	6.6
SUKI run 2	7	180,000	4.7

Table 6: Parameters for the closed training.

For the 3^{rd} edition of the task, we used the HeLI-method without any modifications and the first run of the 4^{th} edition was run with an identical system. This year the Peruvian Spanish replaced the Mexican Spanish. It seems that it is more easily distinguished, at least with the HeLI method, from the Argentinian or Peninsular varieties, as the average F1-score for the Spanish varieties rose from last year's 0.80 to 0.86. Also the inclusion of the languages using the Arabic script helped to raise the overall average F1-score from 0.888 to 0.905.

6 Discussion

After this year's shared task we also looked into the backoff function of the HeLI method and calculated how often each of the *n*-gram lengths were used with the test set. These calculations can be seen in Table 7.

Number of words	*n*
176,635	8
57,252	7
56,361	6
56,243	5
88,054	4
27,975	3
3	2
0	1

Table 7: Number of words identified with each length of *n*-gram.

Table 8 shows the number of words of each length after removing non-alphabetic characters and adding extra space before and after the word. When comparing the two tables it seems that the backoff function was used only with a small fraction of words.

7 Conclusions

Using the loglike function with the actual test set improved the result much more than with the development set. The reduction on the error rate of the accuracy was 4.8%, which was around ten

Number of words	length
60,108	¿10
33,243	10
41,731	9
46,448	8
56,229	7
54,611	6
54,912	5
87,385	4
27,856	3

Table 8: Number of words of each length.

times higher than with the development set. In the future, we will be making further experiments trying to introduce discriminating features into the HeLI method. As it is now, it is still a generative method, not relying on finding discriminating features between languages.

Acknowledgments

We would like to thank Kimmo Koskenniemi for many valuable discussions and comments. This research was made possible by funding from the Kone Foundation Language Programme.

References

Adrien Barbaresi. 2016. An Unsupervised Morphological Criterion for Discriminating Similar Languages. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 212–220, Osaka, Japan.

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 4th VarDial Workshop*, Valencia, Spain.

Ralf D. Brown. 2012. Finding and identifying text in 900+ languages. *Digital Investigation*, 9:S34–S43.

Ralf D. Brown. 2014. Non-linear mapping for improved identification of 1300+ languages. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 627–632, Doha, Qatar.

Yew Choong Chew, Yoshiki Mikami, Chandrajith Ashuboda Marasinghe, and S. Turrance Nandasara. 2009. Optimizing n-gram order of an n-gram based language identification algorithm for 68 written languages. *International Journal on Advances in ICT for Emerging Regions (ICTer)*, 2(2):21–28.

Cyril Goutte, Serge Léger, Shervin Malmasi, and Marcos Zampieri. 2016. Discriminating Similar Languages: Evaluations and Explorations. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016).*

Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. 2015. 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. 2010. Tekstin kielen automaattinen tunnistaminen. Master's thesis, University of Helsinki, Helsinki, Finland.

Nikola Ljubešic and Denis Kranjcic. 2014. Discriminating between very similar languages among twitter users. In *Proceedings of the Ninth Language Technologies Conference*, pages 90–94, Ljubljana, Slovenia.

Nikola Ljubešic and Denis Kranjcic. 2015. Discriminating between closely related languages on twitter. *Informatica*, 39.

Nikola Ljubešic, Nives Mikelic, and Damir Boras. 2007. Language indentification: How to distinguish similar languages? In *Information Technology Interfaces, 2007. ITI 2007. 29th International Conference on*, pages 541–546, Cavtat/Dubrovnik, Croatia.

Marco Lui. 2014. *Generalized language identification.* Ph.D. thesis, The University of Melbourne.

Wolfgang Maier and Carlos Gómez-Rodríguez. 2014. Language variety identification in spanish tweets. In *Proceedings of the EMNLP'2014 Workshop: Language Technology for Closely Related Languages and Language Variants (LT4CloseLang 2014)*, pages 25–35, Doha, Qatar.

Shervin Malmasi, Eshrag Refaee, and Mark Dras. 2015. Arabic dialect identification using a parallel multidialectal corpus. In *Proceedings of the 14th Conference of the Pacific Association for Computational Linguistics, PACLING'15*, pages 209–217, Bali, Indonesia.

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.

Seppo Mustonen. 1965. Multiple discriminant analysis in linguistic problems. *Statistical Methods in Linguistics*, 4:37–44.

Bali Ranaivo-Malançon. 2006. Automatic identification of close languages–case study: Malay and indonesian. *ECTI Transaction on Computer and Information Technology*, 2(2):126–133.

Liling Tan, Marcos Zampieri, Nikola Ljubešic, and Jörg Tiedemann. 2014. Merging comparable data sources for the discrimination of similar languages: The dsl corpus collection. In *Proceedings of the 7th Workshop on Building and Using Comparable Corpora*, Reykjavik.

Jörg Tiedemann and Nikola Ljubešic. 2012. Efficient discrimination between closely related languages. In *Proceedings of COLING 2012*, pages 2619–2634, Mumbai.

Marcos Zampieri and Binyam Gebrekidan Gebre. 2012. Automatic identification of language varieties: The case of portuguese. In *11th Conference on Natural Language Processing (KONVENS) - Empirical Methods in Natural Language Processing - Proceedings of the Conference on Natural Language Processing 2012*, pages 233–237, Vienna.

Marcos Zampieri, Binyam Gebrekidan Gebre, and Sascha Diwersy. 2012. Classifying pluricentric languages: Extending the monolingual model. In *Proceedings of the Fourth Swedish Language Technlogy Conference (SLTC2012)*, pages 79–80, Lund.

Marcos Zampieri, Binyam Gebrekidan Gebre, and Sascha Diwersy. 2013. N-gram language models and pos distribution for the identification of spanish varieties. In *Actes de TALN'2013 : 20e conférence sur le Traitement Automatique des Langues Naturelles*, pages 580–587, Sables d'Olonne.

Marcos Zampieri, Liling Tan, Nikola Ljubešić, and Jörg Tiedemann. 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 (VarDial)*, pages 58–67, Dublin, Ireland.

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. 2013. Using bag-of-words to distinguish similar languages: How efficient are they? In *Computational Intelligence and Informatics (CINTI), 2013 IEEE 14th International Symposium on*, pages 37–41, Budapest.