# Improving Neural Network Performance by Injecting Background Knowledge: Detecting Code-switching and Borrowing in Algerian texts

**Wafia Adouane, Jean-Philippe Bernardy and Simon Dobnik**
Department of Philosophy, Linguistics and Theory of Science (FLoV),
Centre for Linguistic Theory and Studies in Probability (CLASP), University of Gothenburg
{wafia.adouane,jean-philippe.bernardy,simon.dobnik}@gu.se

## Abstract

We explore the effect of injecting background knowledge to different deep neural network (DNN) configurations in order to mitigate the problem of the scarcity of annotated data when applying these models on datasets of low-resourced languages. The background knowledge is encoded in the form of lexicons and pre-trained sub-word embeddings. The DNN models are evaluated on the task of detecting code-switching and borrowing points in non-standardised user-generated Algerian texts. Overall results show that DNNs benefit from adding background knowledge. However, the gain varies between models and categories. The proposed DNN architectures are generic and could be applied to other low-resourced languages.

## 1 Introduction

Recent success of DNNs in various natural language processing (NLP) tasks has attracted attention from the research community attempting to extend their application to new tasks. Nevertheless, the large amount of labelled data required to train DNNs limits their application to new tasks and new languages because it is hard to find large labelled corpora for these domains. The issue is even more severe for low-resourced languages.

Another serious problem with most current NLP approaches and systems is that they are trained on well-edited standardised monolingual corpora, such as the Wall Street Journal, Wikipedia, etc. This could be explained by the fact that for a long time NLP has been influenced by the dominant descriptive linguistic theories affected by the standard language ideology which assumes that natural languages are uniform and monolingual. However, standardisation is not universal (Milroy, 2001), meaning that not all languages are standardised. Therefore, lexical, structural and phonological variation is, for instance, the norm in natural language and not an exception, meaning that well-edited texts do not really reflect the natural usage of natural languages, but only represent formal languages.

The discrepancy between the assumed uniformity of language both in linguistic theory and NLP and their variable nature is accentuated by new technologies, such as social media platforms and messaging services. These new communication platforms have facilitated the proliferation of writing in non-standardised languages on the web, such as colloquial Arabic or what is commonly referred to as dialectal Arabic. This is because in interactive scenarios people usually use spoken-like (colloquial) language or, in multilingual societies where people have access to several linguistic codes at the same time, a mixture of languages/language varieties. Consequently, this new kind of written data has created a serious problem regarding the usability of the existing NLP tools and approaches as they fail to properly process it, even in the case of well-resourced languages.

The contribution of the paper is to explore how to mitigate the problems (i) of the scarcity of annotated data when using DNNs with low-resourced languages, and to what extent can we take advantage of the limited available resources, and (ii) to provide NLP approaches and tools that would be able to deal with non-standardised texts and language-mixing. In particular, for (i) we investigate what are the optimal ways of injecting available background knowledge to different configurations of DNNs in order to improve their performance. For (ii) we take the case of the language used in Algeria as it poses serious challenges for the available NLP approaches and tools.

*Proceedings of The Third Workshop on Computational Approaches to Code-Switching*, pages 20–28
Melbourne, Australia, July 19, 2018. ©2018 Association for Computational Linguistics

It is a low-resourced multilingual colloquial language. We chose the task of a word-level language identification which is a first step towards processing such texts. The task focuses on detecting code-switching and borrowing points in a text which represents the same utterance. Knowing what parts of text belong to what language variety allows to perform better qualitative and quantitative analysis of such texts with other tools.

The paper is organised as follows: in Section 2 we briefly describe the complex linguistic situation in Algeria as a result of a language contact. The section aims to explain the linguistic challenges of processing such texts and motivates our choices based on established sociolinguistic theories. In Section 3 we present our available linguistic resources and different DNN configurations. In Section 4 we describe our experimental setup and analyse the results. Finally, in Section 5 we compare our contribution to previous related work.

## 2 Linguistic Background

In North Africa in general, and in Algeria in particular, intense language contact between various related and unrelated languages has resulted in a complex linguistic situation where several languages are used in a single communicative event. A few cases of language contact have attracted the attention of the linguistic community while the monolingual norm dominates in linguistics. One kind of language contact situation has been described by Ferguson (1959) as *diglossia* which refers to a situation where two linguistic systems coexist in a functional distribution within the same speech community. In another kind of language contact situations, several languages coexist but not in a well-defined functional distribution. This situation is referred to as *bilingualism* (Sayahi, 2014) which could result from either informal contact between coexisting languages like Berber and Arabic, or from formal education where in addition to other language people learn French with varying degrees of competence.

Based on the Fishman's model (Fishman, 1967), North African Arabic, known as Maghrebi Arabic, is classified as a linguistic situation in the speech community characterised by diglossia with bilingualism. The intense language contact between related and unrelated languages has resulted mainly in two widespread linguistic phenomena: code-switching and borrowing. As defined by Poplack and Meechan (1998), code-switching is (ideally) integration of material from one language to another without any phonological, morphological or syntactic integration, whereas borrowing is when material is integrated.

For computational purposes, we focus on *diglossic code-switching* (Sayahi, 2014), which happens between related languages such as switching between Arabic varieties, and *bilingual code-switching*, which happens between unrelated languages such as switching between one Arabic variety and other coexisting language such as Berber, French or English. Regarding borrowing, it is practically not possible to clearly distinguish whether a word in one Arabic variety is integrated into another variety or not because there are no lexicons for Arabic varieties, except for the standard one, and we also do not have access to acoustic representations of words. Based on this, we can practically focus only on *bilingual borrowing* rather than on *diglossic borrowing*.

(1) a. ديري سربيتة صغيرا في طاسة تاع ما او دوبي فيها اسبيجيك وكمديلو بيها نوغمال هاك مداري ندير يريح تم تم

b. Put a small towel in a cup of water and dissolve Aspegic in it and cover him with it, it is what I usually do. He will feel quickly better.

As illustration, (1) is a user-generated utterance which contains words in Modern Standard Arabic (صغيرا، في، فيها), note that the word صغيرا is misspelled and should be spelt like صغيرة, words in local Algerian Arabic ( ديري، تاع، ما، او، دوبي، وكمديلو، بيها، هاك، مداري، ندير، يريح ، تم تم), French words integrated in Arabic (سربيتة، طاسة) , and a French word without integration (نوغمال).

## 3 Linguistic Resources and Models

### 3.1 Linguistic Resources

We use the dataset by (Adouane and Dobnik, 2017) where each word is tagged with a label identifying its category which could be a language/language variety, including local Arabic varieties (ALG), Modern Standard Arabic (MSA), French (FRC), Berber (BER), English (ENG), non-Arabic words integrated in local Arabic or what is referred to as borrowing (BOR), in addition to language independent categories such as named entities (NER), digits (DIG) and interjections (SND). To the best of our knowledge this is the only available labelled dataset for code-switching and borrowing for Algerian. As the labelled dataset is small, we also collected a larger unlabelled dataset from the same sources as the authors of the labelled dataset, and pre-processed them in the same way. Table 1 gives information about the datasets where texts refer to social media texts with an average length of 19 words, words refer to linguistic words excluding other tokens (digits, punctuation, emoticons), and types refer to unique words.

| *Dataset* | *#Texts* | *#Words* | *#Types* |
|---|---|---|---|
| *Labelled* | 10,590 | 213,792 | 57,054 |
| *Unlabelled* | 311,130 | 4,928,827 | 350,759 |

Table 1: Statistics about the datasets.

We also use the lexicons compiled by the authors of the labelled dataset, with further cleaning. The lexicons include lists of inflected words checked manually, one list per category. Words belonging to more than one category are not included. Table 2 gives more information about the sizes of the lexicons.

| *Category* | *ALG* | *MSA* | *FRC* | *BOR* | *NER* | *ENG* | *BER* |
|---|---|---|---|---|---|---|---|
| *#Types* | 42,788 | 94,167 | 3,206 | 3,509 | 1,945 | 165 | 21,789 |

Table 2: Statistics about the lexicons.

### 3.2 Models

We approach the task of detecting code-switching and borrowing points in text as a sequence tagging problem where the aim is to assign a tag to each word in the text depending on its context. We use two DNN architectures, namely Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) with different configurations summarised in Figure 1.

The first option is to use an RNN to map character embeddings to tags directly. Alternatively, we can use word embeddings. Word embedding can be any combination of (a) fixed lexicon information (b) fasttext embeddings (c) a custom CNN built from character embeddings. The word embeddings can be mapped to tags using either an RNN or a CNN, or a simple dense layer with softmax activation.

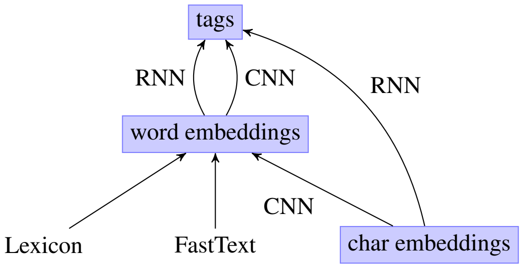


Figure 1: A summary of possible tagging models.

Except for the pure Lexicon-based model, all other models have access to characters and thus to the internal structure of words (phoneme and morphemes), which we expect to be predictive of a particular variety. All models are trained end-to-end, except for the fasttext embeddings and the lexicon. We report only the configurations of models which give the best performance, with the fine-tuned parameters, namely the number of units for each RNN layer, dropout rate, the number of features and the filter size for each CNN layer. The parameters are fine-tuned on a separate development set containing 1,000 texts (13,771 tokens).

#### 3.2.1 Character-level RNN

The character-level RNN is composed of two LSTM layers of 400 units each, with a dropout of 10%, followed by a dense layer with softmax activation. Due to the nature of RNNs, the network assigns one language variant per input symbol, and thus per character — but the task is to predict a tag for each word. To deal with this limitation, we consider only the tag associated with the last character of a word.

#### 3.2.2 Word-level RNN

The word-level RNN is composed of a standard LSTM layer with 400 units with a dropout of 10%, followed by a dense layer.

#### 3.2.3 Character-level CNN

The character-level CNN is composed of two convolution layers with 60 features with a filter size 5, with a relu activation and a dropout of 10%, followed by max pooling in the temporal dimension.

### 3.2.4 Word-level CNN

The word-level CNN is composed of two convolution layers with a filter size 3, with a relu activation and a dropout of 10%, followed by a dense layer with softmax activation. The first layer uses 100 features and the second 60 features.

### 3.2.5 Lexicon-based Model

In order to take advantage of the available lexicons, Table 2, we represent their words as one-hot encoding vector, which we refer to as lexicon embeddings. The lexicon-based model is composed of the lexicon embeddings followed by two convolution layers with a filter size 3, with a relu activation and a dropout of 10%, followed by a dense layer with softmax activation. The first layer uses 100 features and the second 60 features.

### 3.2.6 FastText-based Model

In order to take advantage of the unlabelled dataset, Table 1, containing a high level of misspellings and spelling variation, we assume that word embeddings that are based on sub-word information capture spelling variation and morphological information better than the embeddings that take word as a unit. For this purpose we use FastText library designed to train word embeddings where a word is represented as the sum of its sub-strings (Bojanowski et al., 2016). We created five fasttext embeddings trained on the unlabelled dataset with different parameters. We found that the optimal parameters are: word vector dimension of 300, and the range of the size of the sub-strings representing a word between 3 and 6 characters, with a context size of 5 words, trained on 20 epochs. The FastText-based model is composed of the fasttext embeddings followed by two convolution layers with filter size 3, with a relu activation and a dropout of 10%, followed by a dense layer with softmax activation. The first layer uses 100 features and the second 60 features.

## 4 Experimental Setup and Results

All models and configurations are evaluated under the same conditions using 10-fold cross-validation on the labelled dataset. As a baseline we take an existing system (Adouane and Dobnik, 2017), a classification-based system which uses a chain of additional back-off strategies which involve lexicons, linguistic rules, and finally the selection of the most frequent category. We refer to this system as the baseline.

First, we train the RNN and CNN models only on the labelled data (supervised learning) without any background knowledge. We also examine the effect of the FastText-based and the Lexicon-based models separately to quantify the contribution of each. Then we combine both models to optimise their performance. Second, in order to take advantage of all available linguistic resources, we add to each of the RNN and the CNN models background knowledge in the form of (i) lexicon embeddings; (ii) fasttext embeddings; (iii) a combination of both lexicon and fasttext embeddings; and (iv) bootstrap the unlabelled dataset with the baseline system and train the best performing DNN model on it to investigate whether bootstrapping improves its performance.

All results are reported as the average performance of the 10-fold cross-validation for each model at epoch 100 using the parameters mentioned earlier. For short, we use FastText to refer to the FastText-based model and fasttext to refer to the fasttext embeddings, Lexicon to refer to the Lexicon-based model and lexicon to refer to the lexicon embeddings.

### 4.1 Models without Background Knowledge

In Table 3 we report the average error rate of the experiments without background knowledge for only the best performing RNN, CNN, Lexicon, and FastText models.

| | Model | ER (%) |
|---|---|---|
| 1 | Char-level RNN | 13.38 |
| 2 | Char-level CNN | **8.18** |
| 3 | FastText | 16.46 |
| 4 | Lexicon | 20.62 |
| 5 | FastText + Lexicon | 9.21 |
| 6 | Baseline | 9.52 |

Table 3: Average error rate of the models without background knowledge.

Results show that the baseline (6) outperforms the Char-level RNN (1), FastText (3) and Lexicon (4) models. However, the baseline is outperformed by the Char-level CNN model (2) with 1.34% error reduction. Combining FastText and Lexicon in one model (5) performs much better than using each model separately, and slightly outperforms the baseline by 0.31% error reduction.

In Figure 2 we report the average performance of each model per category, measured as precision, recall, f-score and loss. Notice that we do not report the loss for the baseline because of the way the system was designed. The results show that the baseline system performs better on the majority categories, ALG and MSA, with an average f-score of 91.91 and 90.44 respectively as well as on non-linguistic categories like DIG and SND with an average f-score of 97.17 and 93.88 respectively.

However, the baseline system performs less well on the minority categories, BER and FRC with an average f-score of 80.41 and 80.31 respectively, and performs even worse on NER and BOR with an average f-score of 72.55 and 64.70 respectively. It performs the worst on ENG with an average f-score of 49.45. Regarding the minority categories, precision is high on BER (94.51%), BOR (93.61%), FRC (92.97%) and lower on NER (88.20%) and ENG (71.41%). However the recall is low on all categories BER (72.76%), FRC (70.70%), NER (61.74%) and the lowest on BOR (49.44%), and ENG (39.37%).

The error analysis of the baseline system shows that the system is mostly confused between related language varieties like ALG-MSA as they share a lot of words, as well as between varieties that share lexically ambiguous words like FRC-ALG, BOR-ALG, FRC-BOR, NER-ALG, BER-ALG. Several words were neither seen in the training data nor were they covered by the available lexicons which, given that the unknown words are tagged as ALG, leads to confusions such as ENG-ALG, NER-ALG, BER-ALG, BOR-ALG, and FRC-ALG.

(2) a. بويسك قالو غادي يقطعو لما سمانة لجاية راني شريت فاردو **ما** فيه ٢٠ **قرعة سعيدة** تاع ليتر ونص

b. Since they said that they will cut water next week, I have bought a load of 20 bottles of Saida of 1.5 litre.

The MSA-NER confusion is mainly caused by the fact that many NERs are simply common nouns in MSA. For instance, سعيدة could be an adjective in the feminine form in MSA meaning *happy*, or a feminine proper name, or something else. In the context of example (2) it is NER as it refers to the name of a product. The word ما means *water* in ALG, but it is also used as a negation particle in MSA and frequently in ALG, a relative pronoun in MSA, and a noun meaning *mother* in ALG. Likewise قرعة means *bottle* in ALG, but it also means *contest* or *competition* in MSA.

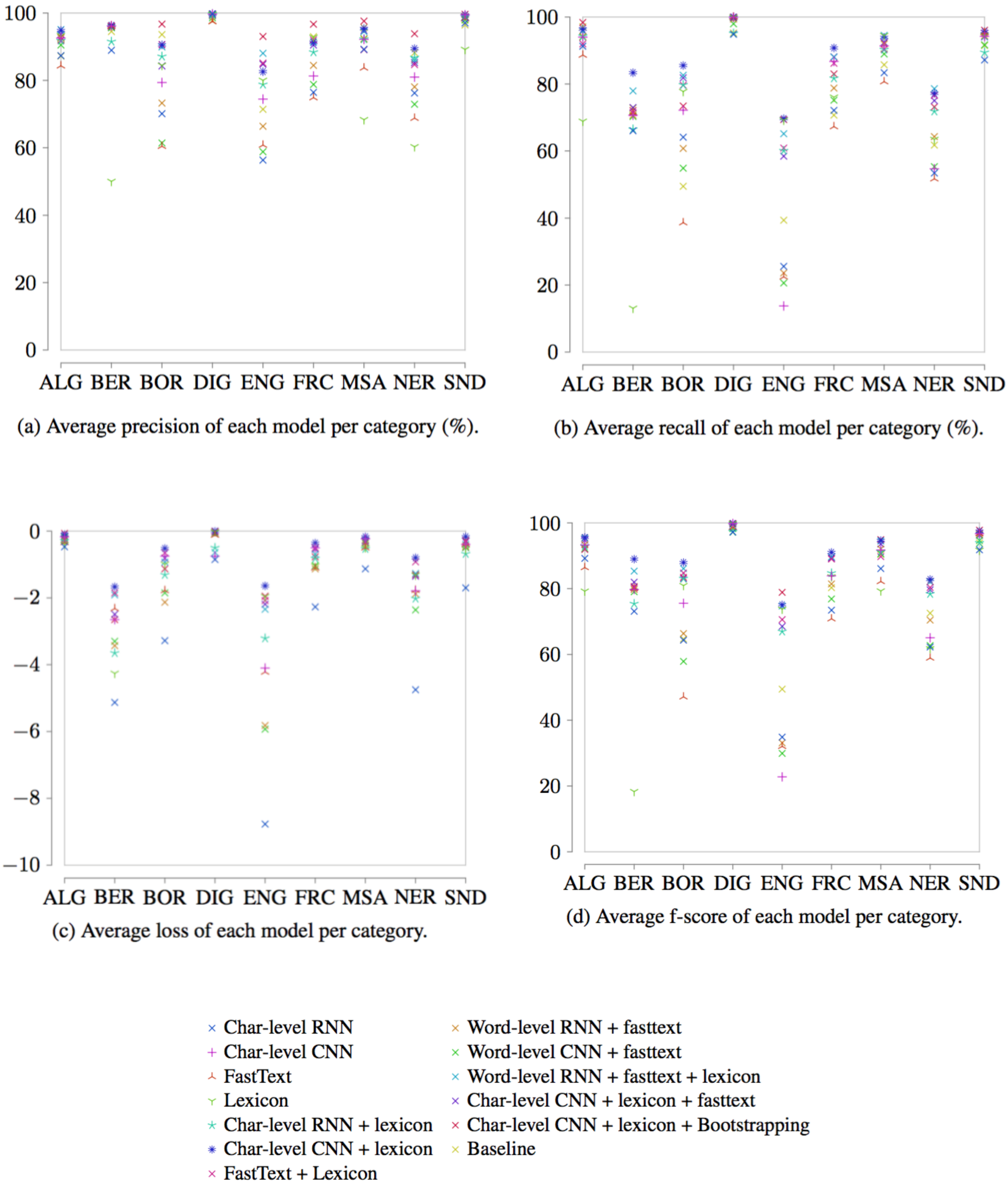
The f-score and precision of the Char-level RNN model is lower from the baseline on all categories, and the recall is better on BOR 64.11% compared to only 49.44% on the baseline, and FRC 72.12% compared to 70.70% respectively. ENG, BER, NER, BOR and FRC are the hardest categories to identify with the following respective loss values: -9.56, -6.72, -3.89, -3.57, -2.80, and all categories are confused with ALG, the majority class.

The f-score of the Char-level CNN model is better on SND, MSA, FRC, DIG, BOR, ALG compared to the baseline, but it performs worse on NER, ENG, BER. This could be contributed by the worse recall on these categories which follows the same trend as the f-score. However, in terms of precision, the Char-level CNN model performs better on ALG, BER, ENG and SND and worse on the remaining categories, with the same kind of confusions as the baseline.

The f-score of the FastText model is low on all categories compared to the baseline. The same holds for recall and precision except on BER where the precision is better 96.18% compared to 94.51% on the baseline. The model produces the same kind of errors as the previous models, but which are most similar to the Char-level CNN model.

Compared to the baseline, the Lexicon model performs better in terms of the f-score on BOR (80.94 compared to 64.70), ENG (73.72 compared to 49.45), and FRC (83.60 compared to 80.30). However it performs significantly worse on BER (18.31 compared to 80.41). This is likely because of the limited coverage of the lexicons. The results also indicate the bias of the lexicons to those categories that are more difficult to distinguish automatically. On the other hand, in terms of the recall, the Lexicon model outperforms the baseline on all categories, except on ALG. In terms of the precision, it is only better on ALG and ENG. The model makes similar errors as the FastText model, only more frequently.

Combining FastText and Lexicon models has a positive effect as the f-score, recall and precision increase on all categories, mainly on BOR (f-score of 47.10 to 84.74), ENG (F-score of 32.05 to 70.59) and NER (f-score of 58.90 to 80.35). The



(a) Average precision of each model per category (%).

(b) Average recall of each model per category (%).

(c) Average loss of each model per category.

(d) Average f-score of each model per category.

Figure 2: Average performance of each model per category.

combined model makes the same errors as previous models but less frequently.

Overall, the results in this section show that a simple Char-level CNN model outperforms the more complicated baseline system which uses a back-off strategy and extra resources. However the Char-level CNN model performs worse on the minority classes, particularly on NER, ENG and BER. On the other hand, the other models perform better on the minority classes in terms of recall, but they perform worse on the remaining categories because of the limited coverage of the lexicons or because of lexical ambiguity. This means that the performance of these models is in complementary distribution. We will explore this observation in the following section.

### 4.2 Models with Background Knowledge

One possible improvement of the models in Section 4.1 is to inject information from the lexicons and the knowledge encoded in the fasttext to the DNN models. In Table 4 we report the average error rate of only the best performing experiments combining different models and resources.

| | Model | ER (%) |
|---|---|---|
| 1 | Char-level RNN + lexicon | 8.27 |
| 2 | Word-level RNN + fasttext | 8.20 |
| 3 | Word-level RNN + fasttext + lexicon | **5,34** |
| 4 | Char-level CNN + lexicon | **5.18** |
| 5 | Word-level CNN + fasttext | 9.75 |
| 6 | Char-level CNN + lexicon + fasttext | 6.23 |
| 7 | Char-level CNN + lexicon + Bootstrapping | 5.23 |
| 8 | Baseline | 9.52 |

Table 4: Average error rate of the models with background knowledge.

The results show that RNN models (with original error rate of 13.38% for Char-level RNN) benefit from both adding the lexicon (1) and the fasttext (2). The gain is even higher when combining both with the Word-level RNN (3). The CNN models behave differently when adding lexicon and fasttext. The Char-level CNN (4) performs best with the lexicon with 3% error reduction. The Word-level CNN (5) performs worse with fasttext compared to basic Char-level CNN introducing a 1.57% increase in the error rate (Table 3). Also the Char-level CNN (6) does not benefit from combining lexicon and fasttext. It appears that the latter introduces noise that CNN is sensitive to. Likewise, additional bootstrapped training data does not help the otherwise best performing Char-level CNN + lexicon model (7). This may be also explained by the additional noise in the bootstrapped data.

Figure 2 indicates that adding lexicon information has a positive effect on the overall performance of the RNN models. The gain from the lexicons is noticeable on all categories where precision, recall and f-score increase, most importantly on BER, BOR, ENG, FRC and NER. The same kind of errors are present as with the previous models but fewer in number. For instance the number of errors between ALG-MSA drops from 1,077 to 724, and between FRC-ALG from 104 to 64.

Adding lexicon information to the Char-level CNN model boosts its overall performance over models not using lexicons. All the categories benefit from the lexicon information and their f-score, recall and precision increase, most importantly on the minority categories such as ENG, with the same errors but less frequent. However, adding fasttext does not improve the performance of the Word-level CNN model. Its average f-score decreases on all categories except on ENG where it increases from 22.76 to 29.91.

Compared with the Char-level CNN + lexicon model, adding fasttext to Char-level CNN does not have the same positive effect. The only significant gain is an increase in precision on ENG from 82.59% to 84.79%. Char-level CNN + fasttext + lexicon model performs better than the FastText + Lexicon model. It seems that fasttext does not help the CNN model.

On the other hand, adding fasttext to an RNN boosts its performance. The error rate drops to 13.38% (Char-level RNN) and 8.20% (Word-level RNN). While the precision of each category improves, the recall drops on both BOR and ENG categories, by 3.35% and 1.97% respectively. The f-score increases on all categories except on ENG where it drops by 1.76%.

Examining the effect of lexicon and fasttext on the RNN models, we find that the precision on the minority categories, chiefly BOR, ENG, FRC, NER is higher when adding lexicon (87.10%, 78.77%, 88.36%, and 86.77%) compared to when adding fasttext (73.26%, 66.37%, 84.45% and 78.07%), but the precision on BER is better when adding the fasttext (96.18% compared to 91.51%). The same trend is observed for recall where BER is the only category that benefits from fasttext compared to lexicon (70.65% compared to 66.47%). ENG is the category which is most negatively effected when adding fasttext with a drastic decrease of 36.45% (23.62% with fasttext and 60.07% with lexicon), followed by BOR with 18.98% decrease, and NER with 7.54% decrease. The f-scores have the same pattern as the recall.

A gain of adding lexicon to the Word-level RNN + fasttext model is observed on all categories. While precision increases on all categories, for example on ENG from 78.77% without the lexicon to 88.04% with the lexicon, it slightly decreases for NER from 86.77% to 85.97% and SND from 99.00% to 98.86%. The recall and f-score increase on all categories.

The gain from using the bootstrapped data is mainly reflected in an increase in precision on the minority categories such as ENG, BOR, FRC and NER (93.04%, 96.71%, 96.68% and 93.85% compared to 82.60%, 90.56%, 91.31% and 89.43% respectively without using the bootstrapped data). In terms of recall, the bootstrapped data only boosts ALG and SND categories. The f-scores of the model trained without the bootstrapped data are better on all categories. The insignificant effect of the bootstrapped data could be attributed to the additional noise introduced by the baseline system.

## 5 Related Work

The emerging digitised multilingual data that followed the introduction of new technologies and communication services has attracted attention of the NLP research community in terms of how to process such linguistic data that resulted from language contact between several related and unrelated languages, for example in detection of code-switching where mainly traditional sequence labelling methods are used for Bengali-English-Hindi (Barman et al., 2014a), Nepali-English (Barman et al., 2014b), Spanish-English and MSA-Egyptian Arabic (Diab et al., 2016), MSA-Moroccan Arabic (Samih and Maier, 2016), MSA-Algerian Arabic-Berber-French-English (Adouane and Dobnik, 2017), etc.

The work most closely related to ours is described in (Samih et al., 2016) who used a supervised LSTM-RNN model combined with Conditional Random Fields to detect switching points between related languages (MSA - Egyptian Arabic) trained on a small dataset from Twitter. However, the system was only evaluated on the majority categories. Similarly, Kocmi and Bojar (2017) proposed a supervised bidirectional LSTM-RNN trained on artificially created multilingual edited texts. These does not fully reflect all the complexities of real linguistic use in a multilingual scenario.

Adouane et al. (2018) propose a character-level GRU-RNN on the same task as described here backed by the available unlabelled data. They report that their supervised RNN model performs the best on labels with more representative samples. Adding neural language model that was pre-trained on noisy unlabelled data does not help, but bootstrapping the unlabelled data with another system improves the performance of all their systems. In this work we use different DNN architectures (RNNs and CNNs), and we aim to examine the behaviour of each model when injecting background knowledge in the form of encoded information from the available lexicons and a pre-trained sub-word embeddings from unlabelled data. Our goal is to take advantage of the available NLP resources, with as little processing as possible to mitigate the problem of scarce annotated data.

## 6 Conclusion

We have presented DNN models for detecting code-switching and borrowing for an under-resourced language. We investigated how to improve these models by injecting background knowledge in the form of lexicons and/or pre-trained sub-word embeddings trained on an unlabelled corpus, thus taking advantage of the scarce NLP resources currently available. The results show that the models behave differently for each category of added knowledge. While adding information from the lexicons markedly improves the performance of all models, adding knowledge in the form of pre-trained sub-word embeddings improves the RNN model more than the CNN model. Bootstrapping does not bring a significant overall contribution to performance of our models which is surprising given the previous reports in the literature. However, it does boost precision of the minority categories. One future direction worth exploring is how to deal with the problem of misspellings and spelling variations to reduce the irregularities in non-standardised user-generated data as this appears to have a strong effect on the performance of RNN and CNN models.

## Acknowledgement

The research reported in this paper was supported by a grant from the Swedish Research Council (VR project 2014-39) for the establishment of the Centre for Linguistic Theory and Studies in Probability (CLASP) at the University of Gothenburg.

## References

Wafia Adouane and Simon Dobnik. 2017. *Identification of Languages in Algerian Arabic Multilingual Documents*. Association for Computational Linguistics.

Wafia Adouane, Simon Dobnik, Jean-Philippe Bernardy, and Nasredine Semmar. 2018. *A Comparison of Character Neural Language Model and Bootstrapping for Language Identification in Multilingual Noisy Texts*. In Proceedings of the Second Workshop on Subword and Character Level Models in NLP, New Orleans, Louisiana USA.

Utsab Barman, Amitava Das, Joachim Wagner, and Jennifer Foster. 2014a. Code-mixing: A challenge for Language Identification in the Language of Social Media. In *In Proceedings of the First Workshop on Computational Approaches to Code-Switching*.

Utsab Barman, Joachim Wagner, Grzegorz Chrupala, and Jennifer Foster. 2014b. Dcu-uvt: Word-Level Language Classification with Code-Mixed Data.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information. *arXiv preprint arXiv:1607.04606*.

Mona Diab, Pascale Fung, Mahmoud Ghoneim, Julia Hirschberg, and Thamar Solorio. 2016. *Proceedings of the Second Workshop on Computational Approaches to Code Switching*. Association for Computational Linguistics, Austin, Texas.

Charles Albert Ferguson. 1959. Diglossia. *WORD, Routledge*, 15(2):325–340.

Joshua Aaron Fishman. 1967. Bilingualism with and without Diglossia; Diglossia with and without Bilingualism. *Journal of Social Issues*, 23:29 – 38.

Tom Kocmi and Ondřej Bojar. 2017. Lanidenn: Multilingual Language Identification on Text Stream. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 927–936. Association for Computational Linguistics.

James Milroy. 2001. Language Ideologies and the Consequence of Standardization. *Journal of Sociolinguistics*, 5:530 – 555.

Shana Poplack and Marjory Meechan. 1998. How Languages Fit Together in Codemixing. *The International Journal of Bilingualism*, 2(2):127–138.

Younes Samih, Suraj Maharjan, Mohammed Attia, Laura Kallmeyer, and Thamar Solorio. 2016. Multilingual Code-switching Identification via LSTM Recurrent Neural Networks. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 50–59.

Younes Samih and Wolfgang Maier. 2016. Detecting Code-Switching in Moroccan Arabic. In *Proceedings of SocialNLP @ IJCAI-2016*.

Lotfi Sayahi. 2014. *Diglossia and Language Contact: Language Variation and Change in North Africa*. Cambridge Approaches to Language Contact. Cambridge University Press.