Neural Models for Detecting Binary Semantic Textual Similarity for Algerian and MSA

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 extent to which neural networks can learn to identify semantically equivalent sentences from a small variable dataset using an end-to-end training. We collect a new noisy non-standardised user-generated Algerian (ALG) dataset and also translate it to Modern Standard Arabic (MSA) which serves as its regularised counterpart. We compare the performance of various models on both datasets and report the best performing configurations. The results show that relatively simple models composed of 2 LSTM layers outperform by far other more sophisticated attention-based architectures, for both ALG and MSA datasets.

1 Introduction

Detecting Semantic Textual Similarity (STS) aims to predict a relationship between a pair of sentences based on a semantic similarity score. It is a well-established problem (Agirre et al., 2012) which deals with text comprehension and which has been framed and tackled differently (Beltagy et al., 2013, 2014). In this work we focus on deep learning approach. For example, Baudis and Šedivý (2016) frame the problem as a sentence-pair scoring using binary or graded scores indicating the degree to which a pair of sentences are related.

Solutions to detecting semantic similarity benefit from the recent success of neural models applied to NLP and have achieved new state-of-the-art performance (Parikh et al., 2016; Chen et al., 2017). However, so far it has been explored only on fairly large well-edited labelled data in English. This paper explores a largely unexplored question which concerns the application of neural models to detect binary STS from small labelled datasets. We take the case of the language used in Algeria (ALG) which is an under-resourced language with several linguistic challenges. ALG is a collection of local colloquial varieties with a heavy use of code-switching between different languages and language varieties including Modern Standard Arabic (MSA), non-standardised local colloquial Arabic, and other languages like French and Berber, all written in Arabic script normally without the vowels.

ALG and MSA are two Arabic varieties which differ lexically, morphologically, syntactically, etc., and therefore represent different challenges for NLP. For instance, ALG and MSA share some morphological features, but at the same time the same morphological forms have different meanings. For instance, a verb in the 1$^{\text{st}}$ person singular in ALG is the same 1$^{\text{st}}$ person plural in MSA. The absence of morpho-syntactic analysers for ALG makes it challenging to analyse such texts, especially when ALG is mixed with MSA. Furthermore, this language is not documented, i.e., it does not have lexicons, standardised orthography, and written morpho-syntactic rules describing how words are formed and combined to form larger units. The nonexistence of lexicons to disambiguate the senses of a word based on its language or language variety makes resolving lexical ambiguity challenging for NLP because relying on exact word form matching is misleading.

(1) a. فوت سمانة في **دار** مواليا كي وليت لقيت
وليدي **دار** حالة منداك نهار راجلي **دار** فرايو
ولا جامي اخليني نبات

b. I spent one week at my parents' **house** and when I came back I found that my son **made** a big mess. After that my husband **changed** his opinion and never allowed me to stay over night (at my parents' house).

(2) a. حنا فلمولود نوجدو طعام **لفطور** ونتوما واش

Proceedings of the Fourth Arabic Natural Language Processing Workshop, pages 78–87
Florence, Italy, August 1, 2019. ©2019 Association for Computational Linguistics

غادي تديرو **غدا**

b. In Mawlid we prepare Couscous for **lunch**, and you what will you prepare (for lunch)?

In many cases, while the same word form has several meanings depending on its context, different word forms have the same meaning. As an illustration, consider examples (1) and (2) which are user-generated texts taken from our corpus (Section 3.1.1). In (1), the same word form "**دار**" occurs three times with different meanings: "house", "made", and "changed" respectively. Whereas in (2), the different word forms "**لفطور**" and "**غدا**" mean both "lunch".

We mention these examples to provide a basic background for a better understanding of the challenges faced while processing this kind of real-world data using the current NLP approaches and systems that are designed and trained mainly on well-edited standardised monolingual corpora. We could, for instance, distinguish the meanings of "**دار**" in (1) if we knew that the 1st occurrence is a noun and the two others are verbs. Likewise, if we had a tool to distinguish between ALG and MSA, it were easier to detect the meaning of "**غدا**" as "lunch" in ALG rather than the MSA meaning "tomorrow".

Traditional models for detecting STS cannot be applied on such data because they require existing resources and tools, such as tokeniser, stemmer, PoS tagger, etc. to pre-process the data and extract useful features assuming that the data is correctly spelled (standardised orthography). Thus using deep neural networks (DNNs) is promising because representations can be learned in an unsupervised way. In particular, when trained end-to-end, inputs are mapped directly to the desired outputs without the need to handcraft features. Nevertheless, this learning approach based on pattern matching requires lot of data to learn useful patterns. Besides there are only a few cleaned and labelled textual corpora available for some languages and creating new ones is labour intensive.

Our contributions are as follows. (i) We introduce a newly built (small) ALG dataset for STS. (ii) We compare the performance of different DNN configurations on this dataset, namely: various combinations of Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), pre-training of embeddings, including a replication of two new state-of-the art attention models. (iii) We test whether increasing the dataset size helps. (iv) We test whether language regularisation helps. For this purpose, we run the same experiments on a regularised and comparable MSA translation of the ALG dataset.

The paper is structured as follows. In Section 2, we briefly review some STS applications. In Section 3, we describe our experimental setup including data and models. In Section 4, we discuss the results and conclude with our future plans in Section 5.

2 Related Work

Diverse techniques and formalisms have been used to deal with various semantic-related tasks. Among others, machine learning has been applied to detect semantic textual relatedness such as Textual Entailment (TE) (Nielsen et al., 2009), STS (Agirrea et al., 2016), Paraphrase Identification (PI) (Liang et al., 2016), etc. Earlier systems use a combination of various handcrafted features and are trained on relatively small datasets. For example, Dey et al. (2016) uses Support Vector Machines with a set of lexical, syntactic, semantic and pragmatic features. As discussed earlier, these features are not available from our dataset.

These tasks have recently attracted more attention when DNNs became practical, mainly due to the availability of large labelled datasets such as the Stanford Natural Language Inference corpus (SNLI) containing 570K sentence pairs (Bowman et al., 2015), Sentences Involving Compositional Knowledge (SICK) containing about 10K sentence pairs (Marelli et al., 2014), the Microsoft Research WikiQA Corpus (WIKIQA) containing more than 23K sentence pairs (Yang et al., 2015), the Quora dataset released by Kaggle competition consisting of 400K potential question duplicate pairs[1], and the Microsoft Research Paraphrase (MSRP) consisting of more than 5K sentence pairs (Dolan and Brockett, 2005).

We follow the approach of Baudis and Šedivý (2016) who consider that several tasks dealing with detecting semantic relatedness are technically similar and can be formulated as sentence-pair

[1] Corpus webpage: `https://www.kaggle.com/quora/question-pairs-dataset`

scoring. They propose a generic framework for text comprehension for evaluating and comparing existing systems. Several DNN systems have been proposed. For instance, Mueller and Thyagarajan (2016) propose a siamese recurrent architecture using Manhattan LSTM (MaLSTM) for STS. They use word embeddings supplemented with synonymy information, LSTM and Manhattan distance to compose sentence representations.

Additionally, complex DNN systems with various attention mechanisms have been proposed to deal with more than one semantic similarity task at the same time. For instance, Yin et al. (2015) apply attention to represent mutual influence between the input sentence pairs. Similarly, Parikh et al. (2016) propose the Decomposable Attention Model (DecompAtten) which relies on alignment using neural attention to decompose the task of natural language inference into sub-tasks which are aggregated and used to predict the output. In the same direction, Chen et al. (2017) propose the Enhanced Sequential Inference Model (ESIM) composed of a bidirectional LSTM (BiLSTM) encoder, and a soft alignment which computes attention weights to determine the relevance between two input sentences. Then they use another BiLSTM layer to compose local inference information and aggregate the output by applying average and max pooling, and concatenating all in one vector.

All preceding models involve considerable sophistication of design and sometimes require specific dataset annotation. This is to say they are normally trained on large well-edited and labelled datasets that are available for English but are unavailable for most other languages. Unlike the previous work, we will compare the performance of two presumably best performing architectures to simpler architectures similar to MaLSTM but with different additional components on a small unedited dataset.

3 Experiment

3.1 Data

3.1.1 ALG STS data

To the best of our knowledge, there is no ready-to-use ALG data for any semantic similarity related task prior to this work. As a basis we use an extended version of the ALG unlabelled dataset (Adouane et al., 2018) which currently contains 408,832 unedited short colloquial texts (more than 6 million words) collected from online discussion forums. For the STS task we created a dataset of 3,000 sentence pairs as follows. We randomly selected 1,000 sentences from the ALG unlabelled data, including various topics and text lengths. We asked two ALG native speakers to produce for each given sentence two more sentences: one which is semantically equivalent and the other can be semantically similar but not equivalent, i.e., it could include the same words or could be about the same topic.

(3) a. لالا ماشي باهية الروز قديم دوكا .
الوردي ما عجبنيش ما هوش الامود .
b. No, it is not beautiful, pink is outdated.
I do not like pink, it is not fashionable.

(4) a. هديت ليما تارت تاع الشوكو .
عجبتني لاتارت تاع الشوكو لي دارتها يما .
b. I offered to my mother a chocolate pie.
I like the chocolate pie that my mother baked.

In (3), the two sentences are semantically equivalent but in (4) the two sentences are roughly about the same topic and include "chocolate pie", "mother" and "I" but some important information differs — like who did what.

The annotators were free to use whatever words as long as the produced sentences sounded natural to them and the above instructions were respected. We provided them with two examples of the desired sentences and explained the difference. We combined all the sentences and created 3,000 unique sentence pairs.

In the second part of dataset creation, we asked three different native speakers to provide a similarity score between 0–5 for each sentence pair following the guidelines used in the SemEval-2016 shared task (Agirrea et al., 2016). Finally, another annotator performed manual checking and majority voting of the annotations.

Because the annotators assigned scores according to their judgement, the resulting data is not balanced in terms of the number of instances per class (0–5) as shown in Table 1. The corpus contains

36,767 words, 7,074 unique words and sentence average length of 5.19 words or 34 characters.

Score	*Interpretation*	*#Pairs*
0	The two sentences are completely dissimilar.	1,550
1	The two sentences are not equivalent, but are on the same topic.	237
2	The two sentences are not equivalent, but share some details.	140
3	The two sentences are roughly equivalent, but some important information differs.	63
4	The two sentences are mostly equivalent, but some unimportant details differ.	16
5	The two sentences are completely equivalent, as they mean the same thing.	994

Table 1: Annotation guidelines and the number of instances in the ALG STS dataset.

We first tried to predict the graded six similarity scores as multi-class STS, but the systems (Section 3.2) only predicted the most frequent classes, namely scores 0 and 5. This behaviour suggests that given the size of the dataset and the number of instances for each class, the classes are not distinguishable enough. Therefore, we re-framed the task as a binary STS: either two sentences are semantically equivalent or not, rather than predicting their graded similarity (Agirre et al., 2015; Xu et al., 2015). To this end, we merged all scores which do not capture semantic equivalence (0 to 4) into a single class, and refer to them as non-equivalent. The remaining score of 5 stands on its own as completely equivalent. The resulting binary labelled data contains 994 equivalent sentence pairs and 2,006 non-equivalent sentence pairs.

3.1.2 MSA STS data

Contrary to ALG, MSA is a well-represented Arabic variety with standardised spelling. We use a large MSA Wikipedia corpus[2] consisting of more than 52 million tokens. We automatically removed all words written in non-Arabic script and punctuation. We refer to this corpus as MSA unlabelled data.

We also created a labelled STS corpus for MSA by commissioning another pair of ALG native speakers to faithfully translate the ALG STS dataset into MSA. They were instructed to keep the order of words and structures as close as possible to the ALG sentences without changing the meaning. We manually checked the quality of the translation, corrected some minor misspellings and checked the corresponding similarity scores (0–5). We proceeded in the same way as for ALG and created a binary MSA STS dataset including equivalent and non-equivalent sentence pairs.

Both binary and multi-class STS MSA datasets have the same number of sentence pairs as their ALG corresponding datasets. However, the MSA datasets have a smaller vocabulary, consisting of only 5,527 unique words from a total of 37,832 words. The average sentence length is 6.84 words or 33.26 characters. The difference in the vocabulary size is mainly due to misspellings and spelling variations in the ALG corpus: it is non-standardised language. Yet both ALG and MSA datasets have relatively short sentences and they are about the same topics since one is a translation of the other.

3.2 Models

All models have the same basic structure. They consist of two identical siamese networks, one for each input sentence as shown in Figure 1. The main differences between the models are in the embeddings, the sentence encoder, the distance measure, and the objective function for the final prediction.

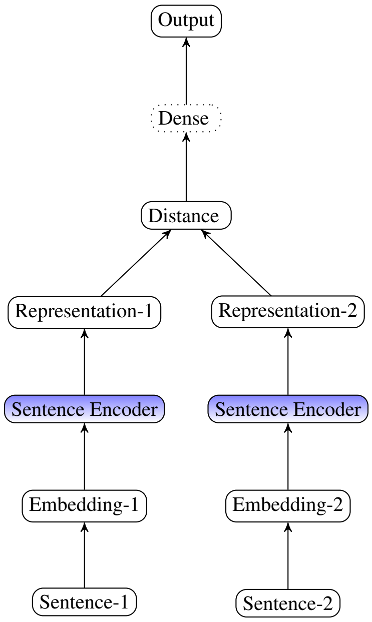

Figure 1: Siamese network architecture. The trained parameters are shared between the left (1) and right (2) part of the network.

[2] The MSA corpus was downloaded from: http://goo.gl/d7pxZb.

3.2.1 Embeddings

We use two kinds of embedding layers. First, an embedding layer trained only on the training data based either on characters or words, initialised either with a uniform or a normal distribution. We refer to these embeddings as trainable as a contrast to pre-trained embeddings. Second, we pre-trained a word2vec and FastText embeddings on the larger unlabelled data mentioned in Section 3.1, using the Gensim (Řehůřek and Sojka, 2010) and FastText (Bojanowski et al., 2016) libraries. For word2vec embeddings, we used a context size of 5 words, minimum occurrence of 1 and dimension of 300. For FastText embeddings, we used dimension of 300, range of sub-characters between 3-5 characters, and a context size of 5 words, and training for 200 epochs. The goal of using pre-trained word embeddings is to test whether we can make use of the large unlabelled corpora.[3]

3.2.2 Sentence Encoders

We use either an RNN or a CNN with different configurations to encode each sentence and output a representation for each. The sentence encoders are identical for both sentences and share weights. Here are some of the encoders that we experimented with.

RNN-based encoder consisting of a stack of standard and/or bidirectional LSTM layers with 300 units and a dropout rate of 3%.

CNN-based encoder consisting of a stack of convolution layers with 60 filters of size 5, with a relu activation and a dropout rate of 10%, followed by max pooling with a pool size of 3, followed optionally by a global average pooling and global max pooling multiplied together.

CNN-RNN-based encoder A combination of RNN and CNN encoders where we stack a number of convolution layers with 60 filters of size 5, with a relu activation and a dropout rate of 10%, followed by max pooling with a pool size of 3 and a number of RNN layers (either standard or bidirectional LSTMs).

Attention-based encoder Roughly put, the idea of an attention mechanism is to attend to some parts of an input/output when deriving its representation (Bahdanau et al., 2014). We implement the Decomposable Attention (DecompAtten) and Enhanced Sequential Inference Model (ESIM) models, as described in Section 2.

3.2.3 Distance

The distance component serves to compose the sentence representations. We use standard distances such as Euclidean distance, Manhattan distance, and Cosine similarity.

3.2.4 Dense

Instead of using a distance measure between the sentence representations, we compose the two sentence representations by multiplication (multp), subtraction (subtr), summation (sum), or concatenation (conct) as in the ESIM model. This operation is followed by a dense layer. We indicate that this layer is optional by using a dotted frame in Figure 1. When it is used, we use a sigmoid activation with a binary cross-entropy loss.

Except for the pre-trained embeddings, all models are trained end-to-end for 300 epochs using a batch size of 64 and Adam optimiser with a learning rate of 0.001.

4 Results and Discussion

We randomly selected from the binary ALG STS dataset 250 sentence pairs of each class (equivalent and non-equivalent) as the test set (500 in total), 200 sentence pairs as a development set, and the remaining 2,300 sentence pairs as a training set. Note that balancing the test set is not essential. Likewise, we split the binary MSA STS data by taking the corresponding translations for each instance in the ALG dataset.

The hyper-parameters reported in Section 3.2 were selected based on the reported common values in the literature for similar tasks and fine-tuned on the development set. Moreover, because of the stochastic nature of the neural models [4] where the results vary between each training run, we report the average performance on the test set over 10 training runs for the best performing models trained on both training and development data following (Baudis and Šedivý, 2016; Yin et al., 2015).

In order to increase the size of the training data and to boost the instances of the minority class

[3] The annotated data and the pre-trained embeddings are available from the 1st author.

[4] https://machinelearningmastery.com/randomness-in-machine-learning/

					ALG		MSA	
	Model	Emb	Encoder	Dist	Acc	Acc-aug	Acc	Acc-aug
1	char-RNN	trainable	2-LSTM	multp	55.78	61.84	59.65	67.80
2	char-RNN	trainable	2-LSTM	subtr	70.38	78.56	69.02	71.37
3	word-RNN	trainable	2-LSTM	multp	**85.06**	87.20	**85.19**	86.69
4	word-RNN	trainable	2-LSTM	subtr	73.73	**92.76**	68.90	88.20
5	word-RNN	word2vec	2-LSTM	subtr	71.40	92.51	67.86	**89.46**
6	word-RNN	FastText	2-LSTM	subtr	71.68	92.70	68.06	88.57
7	word-CNN	trainable	1-CNN	sum	50.00	50.00	50.00	50.00
8	DecompAtten	trainable	attention	sum	50.44	53.00	50.02	50.44
9	ESIM	trainable	attention	conct	52.34	52.80	50.34	50.39

Table 2: Average accuracy of the models (%). Acc is accuracy with non-augmented training data and Acc-aug with the augmented training data.

(equivalent sentence pairs), we duplicated equivalent sentence pairs by reversing their order so that each sentence pair appears only once in the same order. This is a standard data augmentation practice used to mitigate the limited availability of labelled training data (Yin et al., 2015; Mueller and Thyagarajan, 2016). The augmented training set contains 3,244 sentence pairs (1,488 equivalent and 1,756 non-equivalent pairs). Because there is no previous work reported for ALG on a similar task, we resort to the binary random guess, namely 50% as a baseline. We report the overall accuracy for the same models with and without the augmented training data, for both ALG and MSA separately. In Table 2, we only report the models that outperform the baseline.

4.1 Binary STS for ALG

Non-augmented data The results show that char-RNNs composed of 2 standard LSTM layers and trainable embedding layer with normal distribution (1) and (2) perform worse than their word-based counterparts (3) and (4). This result contradicts the conclusion that character models are better at modelling morphologically rich languages (Vylomova et al., 2017), and consequently they are better in dealing with misspellings and capturing spelling variations.

The best performance is achieved by a word-based 2-LSTM layer encoder and a trainable embedding layer (3), using multiplication as a distance with an accuracy of 85.06%. Nevertheless, char-RNN performs better with subtraction rather than multiplication as a distance (2). Adding pre-trained embeddings word2vec (5) and FastText (6) to the word-level RNN in (4) decreases the accuracy by 2.33 and 2.05 points respectively. This effect could be caused by the noise in the ALG unlabelled data on which the embeddings were trained.

A 1-layer CNN with no pre-trained embeddings and using summation of the sentence representations as a distance (7) performs the best compared to the other options with CNN encoder but overall it performs quite poorly. Likewise combining 1-CNN and 1-LSTM layers as encoder (not shown in Table 2) does not have an effect over using only 1-CNN layer. The models predict all the test sentence pairs as non-equivalent. In other words, the network could not learn enough to properly distinguish between the two classes.

These results contrast those reported by Kadlec et al. (2015), namely that CNN models perform better with little data compared to RNN models. However, it is hard to quantify what is considered to be small apart from the number of examples. In general, neural models learn useful features when they are trained on enough representative data. That is to say it is not just a question of data size, but it is more about the complexity of the features and the functions that they should learn. In our case, we suspect that the sparsity and the noise in the data is making learning harder for CNN models.

Regarding attention-based encoders, ESIM (9) outperform DecompAtten (8), and both perform slightly better than the baseline. The poor performance of these models with little noisy data could be related to the fact that attending to some parts of a sentence or focusing on surface form similarity is misleading since the same word form can have different meanings and different word forms can have the same meaning, especially that the data does not contain named entities or punctuation or digits which could help alignment.

Augmented data All models benefit from the augmented data, except word-CNN (7) for which the gain is not clear. The performance of the char-

		Equivalent			Non-equivalent		
	Model	Precision (%)	Recall (%)	F-score	Precision (%)	Recall (%)	F-score
1	char-RNN-multp	73.91	53.54	62.10	63.12	80.80	70.88
2	char-RNN-subtr	88.02	66.54	75.78	72.76	90.80	80.78
3	word-RNN-multp	86.96	88.00	87.48	87.85	86.80	87.32
4	word-RNN-subtr	89.67	97.20	93.28	96.94	88.80	92.69
5	word-RNN-word2vec	89.30	96.80	92.90	96.51	88.40	92.28
6	word-RNN-FastText	**90.84**	95.20	92.97	94.96	**90.40**	92.62

Table 3: Average performance of the models per class trained on the ALG augmented data.

RNN (2) shows 8.18 point improvement in accuracy. This result supports the hypothesis that the poor performance of the model trained on the non-augmented data is caused by the small size of the sparse noisy data which makes it hard for the char-RNN to learn useful patterns. Yet the significant improvement of the word-RNN (4) by 19.03 points, indicates that word-RNN suits better our case.

Models with subtraction as a distance benefit the most from the added data. Similar to their behaviour on non-augmented data, adding pre-trained embeddings slightly decreases the performance of the model compared to not adding them. Comparing embeddings, word2vec causes slightly more drop in the performance of word-RNN compared to FastText. Attention-based models benefit also from the added data, but the gain is larger for DecompAtten compared to ESIM.

Looking at the performance of the models for each class shown in Table 3, it is clear that the RNN models are doing quite well for both classes whereas CNN and Attention-based models, not included for space limits, are too biased to the non-equivalent class. Figures in bold are meant to highlight the gain due to pre-trained embeddings.

Error analysis of the word-RNN model (4) shows that 7 equivalent sentence pairs are misclassified as non-equivalent and 28 non-equivalent sentence pairs are misclassified as equivalent. We manually checked the errors and found that most of the non-equivalent pairs misclassified as equivalent have at least one word in common as in example (5) but the words have a different meaning depending on their context. However, distinguishing between word senses is hard because the context is not entirely sufficient. Example (6) is an equivalent pair misclassified as non-equivalent. The common pattern among the misclassified examples is that they have no exact words in overlap. This could explain why attention-based encoders, with some form of alignment, fail to generalise to new instances. Probably there is a bias to the form with one meaning when senses are not sufficiently differentiated.

(5) a. شفت حجا **بيزار** .
سي **بيزار** ما شفتوش .
b. I saw a weird thing.
It is weird that I did not see it.

(6) a. راني **نخم** وقتاش تدخل لبورس .
يادرى هذيك المنحة وينتا **تجي** .
b. I am thinking when the grant will be received.
I wonder when the grant will be paid.

4.2 Binary STS for MSA

We now evaluate the performance of the same DNN configurations on parallel regularised MSA data using the same hyper-parameters as in Section 4.1. The results are reported in Table 2.

Non-augmented data Again, the word-RNN with multiplication (3) performs the best with an accuracy of 85.19%. The char-RNN (1) with the same settings achieves an accuracy of only 59.65%. Using subtraction, the char-RNN (2) slightly outperforms the word-RNN (4), with 69.02% and 68.90% accuracy respectively. Adding FastText (6) and word2vec (5) pre-trained embeddings causes the accuracy of the best word-RNN (4) of 68.90% to decrease slightly to 68.06% and 67.86% respectively. This could be due to the embeddings not distinguishing between the different senses of the same word, i.e., output one vector representation for each word form. Also the large MSA corpus on which the embeddings were trained can have different topical distribution than the MSA STS data. As with the ALG data, CNN (7) and attention-based encoders (8–9) behave the same.

Augmented data Trained on augmented data, models with subtraction yield the best performance compared to multiplication, and word-RNN (4) outperforms char-RNN (2) with 88.20%

		Equivalent			Non-equivalent		
	Model	Precision (%)	Recall (%)	F-score	Precision (%)	Recall (%)	F-score
1	char-RNN-multp	69.86	61.20	65.25	65.48	73.60	69.30
2	char-RNN-subtr	76.35	62.25	68.58	67.92	80.57	73.70
3	word-RNN-multp	87.04	86.00	86.52	86.17	87.20	86.68
4	word-RNN-subtr	85.77	91.60	88.59	90.99	84.80	87.78
5	word-RNN-word2vec	**87.17**	**92.77**	**89.88**	**92.21**	**86.23**	**89.12**
6	word-RNN-FastText	**86.97**	91.16	**89.02**	90.64	**86.23**	**88.38**

Table 4: Average performance of the models per class trained on the MSA augmented data.

and 71.37% accuracy respectively. Unlike when using the ALG data, pre-trained embeddings improve slightly the performance of (4) with 0.37 (6) and 1.26 (5) points gain in the error reduction respectively. The positive effect of the pre-trained models could be due to the fact that more regularities are captured. Training on augmented MSA data does not yield any significant gain over training on non-augmented data for CNN (7) and attention based models (8–9).

In Table 4 we report the performance of each model per class. Due to space limits, we do not include the CNN and attention-based models which are again struggling with the equivalent class and are biased towards the non-equivalent class. The gain from the pre-trained embedding is in bold. The models perform almost the same for both classes but slightly worse than with the ALG data.

Example (7) is a non-equivalent sentence pair misclassified as equivalent, and example (8) is an equivalent pair misclassified as non-equivalent by the word-RNN model (5).

(7) a. الكيكة أنا أيضا جربتها والله روعة وجدت أولادي كلهم لعقوها .
جربتها كم من مرة كانت سامة وحاسدة .

b. I also tried the cake and it was great, I discovered that my kids finished it.
I tested her many times and she was jealous and envious.

(8) a. يا ليتهم يغيرون **المذيعة** هذه .
يا ريت يغيرون لنا هذه **المنشطة** .

b. Wish they change this presenter.
Hope they will replace this presenter.

It is hard to explain why these examples are misclassified, except that there is not enough context to discover the meaning of the words. For instance, in (8) the words in bold "مذيعة" , "منشطة" are synonyms in these two sentences, and the two sentences have two more word overlaps "هذه" and "يغيرون" with the same meaning. This should help classifying the two sentences as equivalent, but it is not the case possibly because their contexts are different.

5 Conclusion and Future Work

We have presented a new STS dataset for ALG user-generated short texts and its MSA translation. We then described the neural network models trained end-to-end with different configurations and compared their performances on a binary STS task. The results show that relatively simple model architectures, composed of two word-based LSTM layers with subtraction as explicit similarity measure used in the training task, suit better our data compared to the other more sophisticated architectures which might require more data to achieve better performance.

We ran the same experiment on the MSA data, but the results were not really different from the ALG data. However, pre-training embeddings performed better with MSA, probably because the language is more regular and knowing some structure ahead helps. The performance improved with more data for the minority class (equivalent sentence pairs) for both ALG and MSA. However, surprisingly the gain of some models with ALG is greater than their gain with MSA. This is probably caused by the noisiness and the sparsity of the data, the linguistic differences between MSA and ALG, the data size, or all these factors together. Further and deeper experiments and analyses are needed for a better understanding of the results.

Overall, the results of the end-to-end training are promising and could be generalised to other related languages or language varieties with the same under-resource settings. As a future work, we want to explore ways to improve the learning capability of neural models from small noisy datasets without handcrafted features, for example by reducing the noise in the colloquial data (ALG) by normalising spelling variation.

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, Jean-Philippe Bernardy, and Simon Dobnik. 2018. Improving Neural Network Performance by Injecting Background Knowledge: Detecting Code-switching and Borrowing in Algerian texts. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 20–28. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393. Association for Computational Linguistics.

Eneko Agirrea, Carmen Baneab, Daniel Cerd, Mona Diabe, Aitor Gonzalez-Agirrea, Rada Mihalceab, German Rigaua, and Janyce Wiebef. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of SemEval-2016*, pages 497–511. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. *Neural Machine Translation by Jointly Learning to Align and Translate*. arXiv:1409.0473.

Petr Baudis and Jan Šedivý. 2016. *Sentence Pair Scoring: Towards Unified Framework for Text Comprehension*. arXiv:1603.06127v4.

Islam Beltagy, Cuong Chau, Gemma Boleda, Dan Garrette, Katrin Erk, and Raymond J. Mooney. 2013. Montague Meets Markov: Deep Semantics with Probabilistic Logical Form. In **SEM@NAACL-HLT*, pages 11–21. Association for Computational Linguistics.

Islam Beltagy, Katrin Erk, and Raymond J. Mooney. 2014. Probabilistic Soft Logic for Semantic Textual Similarity. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1210–1219. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. *Enriching Word Vectors with Subword Information*. arXiv:1607.04606v2.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A Large Annotated Corpus for Learning Natural Language Inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for Natural Language Inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Kuntal Dey, Ritvik Shrivastava, and Saroj Kaushik. 2016. A Paraphrase and Semantic Similarity Detection System for User Generated Short-Text Content on Microblogs. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2880–2890. The COLING 2016 Organizing Committee.

William B. Dolan and Chris Brockett. 2005. Automatically Constructing a Corpus of Sentential Paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Rudolf Kadlec, Martin Schmid, and Jan Kleindienst. 2015. *Improved Deep Learning Baselines for Ubuntu Corpus Dialogs*. arXiv:1510.03753.

Chen Liang, Praveen Paritosh, Vinodh Rajendran, and Kenneth D. Forbus. 2016. Learning Paraphrase Identification with Structural Alignment. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 2859–2865. AAAI Press.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK Cure for the Evaluation of Compositional Distributional Semantic Models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association (ELRA).

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese Recurrent Architectures for Learning Sentence Similarity. In *AAAI*, pages 2786–2792. AAAI Press.

Rodney d. Nielsen, Wayne Ward, and James h. Martin. 2009. Recognizing Entailment in Intelligent Tutoring Systems*. *Nat. Lang. Eng.*, 15(4):479–501.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A Decomposable Attention Model for Natural Language Inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255. Association for Computational Linguistics.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.

Ekaterina Vylomova, Trevor Cohn, Xuanli He, and Gholamreza Haffari. 2017. Word Representation Models for Morphologically Rich Languages in Neural Machine Translation. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 103–108. Association for Computational Linguistics.

Wei Xu, Chris Callison-Burch, and Bill Dolan. 2015. SemEval-2015 Task 1: Paraphrase and Semantic Similarity in Twitter (PIT). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 1–11. Association for Computational Linguistics.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A Challenge Dataset for Open-Domain Question Answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018. Association for Computational Linguistics.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2015. ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs. *Transactions of the Association for Computational Linguistics*.