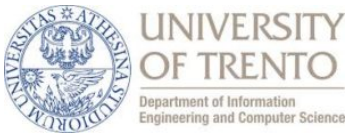
ICNLSP 2019

Proceedings of the First International Workshop on NLP Solutions for Under Resourced Languages (NSURL 2019) co-located with ICNLSP 2019

11–12 September, 2019
University of Trento
Trento, Italy



Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-952148-53-8

Introduction

Welcome to NSURL2019, the First International Workshop on NLP Solutions for Under Resourced Languages (NSURL 2019) co-located with ICNLSP 2019, held on September 11th, 12th 2019, at the University of Trento in Italy. NSURL is an opportunity and a forum for researchers and students to exchange ideas and discuss research and trends in the field of Natural Language Processing and Speech Processing. 26 papers have been submitted to NSURL 2019. 19 of them have been accepted. All the papers have been presented orally. The workshop, indeed, has been an interesting forum for solving NLP problems for low-resourced languages.

The attendance beneficiated from the two keynotes presented at ICNLSP 2019. The first one, entitled "Detecting the fake news before they were even written", presented by Dr. Preslav Nakov from Qatar Computing Research Institute (QCRI), Qatar. The second keynote "One world - seven thousand languages" presented by Prof. Fausto Giunchiglia from University of Trento, Italy. We would like to acknowledge the support provided by University of Trento and Data-Scientia. We would like also to express our gratitude to the organizing and the program committees for the hard and valuable contributions.

Abed Alhakim Freihat, and Mourad Abbas

Trento, September 2019

Organizers:

Chair: Dr. Abed Alhakim Freihat
Co-Chair: Dr. Mourad Abbas

Program Committee:

Mourad Abbas, CRSTDLA, Algeria
Ahmed AbuRa'ed, Universitat P. F. Barcelona, Spain
Abdallah Abushmaes, Mawdoo3 Ltd, Jordan
Abdulmohsen Althubaity, The NCAIBD Research Center-KACST, KSA
Violetta Cavalli-Sforza, Al Akhawayn University, Morocco
Shumile Chabalala, Tshwane University of Technology, South Africa
Abdelrahim Elmadany, Uinversity of Jazan, KSA
Heshaam Faili, University of Tehran, Iran
Mohammad Gharib, University of Florence, Italy
Osama Hamed, University of Duisburg-Essen, Germany
Linda van Huyssteen, Tshwane University of Technology, South Africa
Gabriel Iwasokun, Federal university of Technology Akure, Nigeria
Mohamed Lichouri, CRSTDLA, Algeria
Itani P. Mandende, Tshwane University of Technology, South Africa
Charles Mann, Tshwane University of Technology, South Africa
Maredi I. Mphahlele, Tshwane University of Technology, South Africa
Nandu C Nair, Univeristy of Trento, Italy
Hussein Natsheh, Mawdoo3 Ltd, Jordan
Gabriel Ogunleye, Federal University,Oye-Ekiti, Nigeria
Sunday Ojo, Tshwane University of Technology, South Africa
O. Olugbara, Durban University of Technology, South Africa
Pius A. Owolawi, Tshwane University of Technology, South Africa
Agnieta B. Pretorius, Tshwane University of Technology, South Africa
Adeyanju Sosomi, University of Lagos, Nigeria
Nasrin Taghizadeh, University of Tehran, Iran
Etienne E. Van Wyk, Tshwane University of Technology, South Africa

Organizing committee:

Gabor Bella, Univeristy of Trento
Mattia Fumagalli, Univeristy of Trento
Nandu C Nair, Univeristy of Trento
Olha Vozna, University of Trento

Invited Speakers:

Prof. Fausto Giunchiglia, University of Trento, Italy.
Dr. Preslav Nakov, Qatar Computing Research Institute (QCRI), Qatar.

Invited Talks

Detecting the "Fake News" before they were even written

Preslav Nakov

Given the recent proliferation of disinformation online, there has been also growing research interest in automatically debunking rumors, false claims, and "fake news". A number of fact-checking initiatives have been launched so far, both manual and automatic, but the whole enterprise remains in a state of crisis: by the time a claim is finally fact-checked, it could have reached millions of users, and the harm caused could hardly be undone. An arguably more promising direction is to focus on fact-checking entire news outlets, which can be done in advance. Then, we could fact-check the news before they were even written: by checking how trustworthy the outlets that published them are.

We will show how we do this in the Tanbih news aggregator (http://www.tanbih.org/), which makes users aware of what they are reading. In particular, we develop media profiles that show the general factuality of reporting, the degree of propagandistic content, hyper-partisanship, leading political ideology, general frame of reporting, stance with respect to various claims and topics, as well as audience reach and audience bias in social media.

One world - seven thousand languages

Fausto Giunchiglia

We present a large scale multilingual lexical resource, the Universal Knowledge Core (UKC), which is organized like a Wordnet with, however, a major design difference. In the UKC, the meaning of words is represented not only with synsets, but also using language independent concepts which cluster together the synsets which, in different languages, codify the same meaning. In the UKC, it is concepts and not synsets, as it is the case in the Wordnets, which are connected in a semantic network. The use of language independent concepts allows for the native integrability, analysis and use of any number of languages, with important applications in, e.g., multilingual language processing, reasoning (as needed, for instance, in data and knowledge integration) and image understanding.

Table of Contents

NSURL-2019 Shared Task 8: Semantic Question Similarity in Arabic

Haitham Seelawi
haitham.seelawi@gmail.com

Ahmad Mustafa
Jordan University of Science
and Technology, Jordan
ammustafa@just.edu.jo

Hesham Al-Bataineh
AI Department
Mawdoo3 Ltd
Amman, Jordan
hisham.bataineh@mawdoo3.com

Wael Farhan
AI Department
Mawdoo3 Ltd
Amman, Jordan
wael.farhan@mawdoo3.com

Hussein T. Al-Natsheh
AI Department
Mawdoo3 Ltd
Amman, Jordan
h.natsheh@mawdoo3.com

Abstract

Question semantic similarity (Q2Q) is a challenging task that is very useful in many NLP applications, such as detecting duplicate questions and question answering systems. In this paper, we present the results and findings of the shared task (Semantic Question Similarity in Arabic). The task was organized as part of the first workshop on NLP Solutions for Under Resourced Languages (NSURL 2019) The goal of the task is to predict whether two questions are semantically similar or not, even if they are phrased differently. A total of 9 teams participated in the task. The datasets created for this task are made publicly available to support further research on Arabic Q2Q.

1 Introduction

Semantic Textual Similarity (STS) is a core task in Natural Language Processing and Understanding (NLP/NLU). Simply put, STS is concerned with inferring the similarity in meaning between a pair of sentences. It should be mentioned that there are other levels of granularity for STS such as: Lexical (i.e. single words), full paragraphs or whole documents.

In this paper, we focus on the STS of a question pair (or *Q2Q* Similarity). We assume that if two questions have the same answers, then they are semantically similar. Otherwise, if the answers are different or partially different, then the pair is considered non-equivalent.

STS provides the basis for Question Answering systems (QA). As the name suggests, QA systems are computer systems which can answer questions posed in a natural language form. These questions can be of either factoid or non-factoid nature. Factoid questions can be defined as questions for which a complete answer can be given in 50 bytes or less (a few words) (Soricut and Brill, 2004). These are typically questions that start with who, what, when or where, and have definitive answers. Non-factoid questions, on the other hand, require longer answers. They are mainly instructional or explanatory in nature.

One possible way to build QA systems using STS is having predefined questions along with their answers. When a user asks a question, a ranked list of these questions can be obtained, and based on that list, the best answer can be returned to the user. This method can be used, both, for factoid and non-factoid questions.

One important application to Q2Q is identifying duplicate questions in community question answering platforms (e.g., quora.com). Users may ask questions that might be already asked and answered by the community. Finding these duplicate questions saves the effort and time spent in answering already answered questions. However, detecting duplicate questions is challenging because these questions, although are semantically similar, they might be phrased differently. Moreover, dealing with the Arabic language in Q2Q similarity is challenging due to several factors. Arabic Q2Q datasets are scarce and limited in size. Moreover, the Arabic language is one of the most morphologically rich languages.

In this paper, we present the results and findings of the shared task (Semantic Question Similarity in Arabic). The task was organized as part of the first workshop on NLP Solutions for Under Resourced Languages (NSURL 2019)[1] The goal of the task is to predict whether two questions are similar or not. A total of 9 teams participated in the task. Among them, 4 teams have provided description papers, which are included in the NSURL workshop proceedings.

The rest of this paper is organized as the following. In Section 2, we discuss previously published

[1] http://nsurl.org/

work relating to Q2Q in Arabic. Section 3 provides an overview of the datasets used in the task. Next, in Section 4 we briefly describe the participants and the approaches they propose. Then we discuss the experiments and analyze the results of the competition in Section 5. Finally, we conclude in Section 6.

2 Related Work

Despite its importance and utility in NLP applications, research on STS at the level of sentences and higher, has only picked up steam in the past ten years (Cer et al., 2017). Nonetheless a lot has been accomplished since, but mainly in the English language. In the case of Arabic, there is plenty of room for new research to advance the current state of the art in this regard (Alian and Awajan, 2018) (Nakov et al., 2016). Therefore, most of our review below will focus on methods developed and used in English mainly, which might not be directly applicable to Arabic.

Some of the earliest methods used in the field made extensive use of so-called knowledge-based semantic similarities between words (Majumder et al., 2016). These can be thought of as lexical databases that model the semantic relationships of different words, taking into consideration their different senses. At the center of these databases is the concept of "synsets", which are groups of words that refer to a specific concept. The most popular such database is WordNet (Miller and Fellbaum, 2007). With the assistance of word alignment methods, various meaningful numerical features pertaining to the lexical units comprising a pair of sentences can be obtained from WordNet. Combined with other textual features, such as Part of Speech (POS), and Term Frequency - Inverse Document Frequency (TF-IDF), and fed into strong classifiers, such methods obtain very good results, albeit on closed domains of assessment (Saric et al., 2012; Sogancioglu et al., 2017; Pilehvar et al., 2013). Nonetheless, it can be easily seen that the construction of such databases, is very expensive in terms of human effort.

Semantic relationships can be modeled using another class of methods named Word Vector Representations (WVR). One of the biggest advantages of such methods is that they are typically trained in an unsupervised manner, making their construction very cheap in terms of human annotation. Some of these methods include Word2Vec (Mikolov et al., 2013), Glove (Pennington et al., 2014), ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018). These word representations significantly boost the performance of machine learning algorithms (Mikolov et al., 2013), especially deep learning-based approaches.

One of the earlier and more basic methods of using WVR in STS, consisted in pooling the corresponding dimensions of tokens in a given sentence, using a specific pooling method, such as the average, or the maximum, to obtain a sentence level representation from WVR. The representation of each sentence in the pair would then serve as the input into a classifier or a predefined measure of similarity. One of the obvious advantages of such a method is its simplicity, and that it can be readily used in many classes of machine learning algorithms. However, it is apparent that by using pooling, we are losing all the information about the order of tokens in the original sentences, which definitely matters in defining the meaning of a sentence. Additionally, by using pooling methods, we are assuming that words and sentences can be represented using the same space size, which is a limitation of such a method (Wieting and Kiela, 2019).

One relatively recent advancement in STS, which accounts for the shortcomings of the pooling methods is the Siamese Recurrent Architecture (Mueller and Thyagarajan, 2016). By using two Recursive Neural Networks (RNNs), with shared weights, the pair of sentences are encoded into a higher dimensional space than the WVR used for the constituent tokens. Given the sequential nature of RNNs, this encoding takes into account the order of tokens in each sentence. The encoding is then fed into a feedforward dense neural network, with a value between 0 and 5 to predict the semantic similarity of the pair. One of the advantages of this method when it comes to inference, is that it can be used to produce a sentence level representation, which, with the use of a simple distance matrices, can be used to measure the similarity between two sentences without the need for the feedforward step (Neculoiu et al., 2016). This translates to much higher scalability in industrial applications. Another advantage is that it can be modified to account for errors in spelling (Neculoiu et al., 2016). Nonetheless, a major drawback of this method is that it requires a substantial amount of annotated data for training.

One method which overcome this limitation is Skip-thought Vectors (SV) (Kiros et al., 2015), which learn to embed text at the level of sentences, by training on continuous text (e.g. books and articles) in an unsupervised fashion. The representations can then used as feature inputs with the method of choice to predict the STS score. However, training SV requires very long period of time (it took about one month back in 2015 (Wieting and Kiela, 2019)).

One problem that most sequential deep learning methods suffer from is that the longer the sequence of text to encode is, the less efficient the representation becomes (Olah and Carter, 2016). This problem has been recently tackled by exploiting the attention mechanism in deep learning architectures. With the use of multi-head attention mechanism in constructing sentence embeddings, the state of the art of NLP in many STS dependent tasks has been significantly increased (Lin et al., 2017).

Another recent and novel development pertaining to STS, makes use of conversational data (Yang et al., 2018). The premise here is that sentences that are semantically related, will elicit similar responses in a conversation. However, an obvious shortcoming of such a method is that it is by design geared toward conversational tasks, as opposed to tasks that are factual by nature.

In a new research, (Al-Bataineh et al., 2019) tackles the issue of handling multiple dialects of the same language. The novel approach makes use of deep contextualized word embeddings (Peters et al., 2018) in addition to focus layer (He and Lin, 2016) to overcome out-of-vocabulary introduced by dialectical words.

As it stands now, the state of the art in STS are Universal Sentence Encoders (USE) (Cer et al., 2018). These encoders are trained on a wide variety of data types and tasks (i.e. using different signals such as entailment and SV like signals), with the idea of transfer learning at their heart. Under the hood, USEs can be powered by one of two deep learning architectures; the first is a transformer network, while the other is a deep averaging network. The main difference between these two versions, is that with the former, higher accuracies can be achieved, but with longer training times, whereas for the latter, training is less computationally intensive, at the expense of some accuracy in the final outcome.

Table 1: Mawdoo3 Q2Q dataset statistics.

Set	Similar	Not Similar	total
Train	5,397	6,600	11,997
Test	1,718	1,997	3,715
Total	7,115	8,597	15,712

3 Dataset

Despite the fact that there is a number of public datasets for QA in English language (such as SQuAD (Rajpurkar et al., 2016) and CoQA (Reddy et al., 2018) to name a few, there is a dearth of such datasets in Arabic. Therefore, we have developed a dataset[2] of 15, 712 pairs of questions, that were annotated and verified by an internal team of qualified natural language annotators. Each pair has a ground truth of either "0" (no semantic similarity), or "1" (denoting semantically similar pairs). We have randomly selected 11,997 pairs for training and used the remaining 3,715 for testing. We made sure that the collected data is balanced, where the number of similar question pairs is comparable with the not similar ones. Table 1 shows a detailed statistics of Mawdoo3 Q2Q dataset.

These questions were designed specifically to contain a balanced number of factoid and non-factoid questions. Additionally, great care was taken in assuring that the pairs of questions have varying STS and LS similarity, in a way that mimics the population of questions asked on the internet by Arabic language users. For example:

من هو رئيس الولَايَات المتحدة الأمريكية؟

which translates to "Who is the president of the United States of America?".

Table 1 lists a small sample of the dataset. The dataset consists of 3 fields, i.e. *question1* containing the text for one of the question pairs, *question2* containing the text of the second question, and *label* which is either 1 if question1 and question2 have a similar answer, or 0 if their answers are different. Figure 1 shows a histogram for a number of words per question against frequency. It can be seen that the maximum question length is 15 words and that the distribution of both *question1* and *question2* is almost the same.

[2] https://ai.mawdoo3.com/nsurl-2019-task8

Table 2: Sample of the Mawdoo3 Q2Q dataset. The dataset is composed of three columns. The first two are text fields containing question1 and question2 while the third column shows the label.

question1	question2	label
مَا هي الطرق الصحيحة لِّاعتنَاء بَالحَامل؟	كيف اهتم بطفلي؟	0
مَا طريقة تحضير محشي الكوسَا ؟	من طرق تحضير محشي الكوسَا؟	1
في أي عَام ولد توفيق الحكيم ؟	أين ولد توفيق الحكيم ؟	0
مَا طريقة تحضير اَالمهلبيخّة بجوز الهند؟	كيف احضر المهلبيخّة بجوز الهند؟	1
مَا طريقة تحضير الكيك المحشي بَالكريمة ؟	من طرق تحضير الكريمة؟	0
مَا هي حصوَات المرَارة؟	مَا هي حصَى المرَارة؟	1
كيف احضرَالمصَابيب مع المكشن؟	من طرق تحضير المصَابيب المحشي؟	0
مَا هو الموت؟	مَا أجمل مَا قيل بَالموت؟	0
في أي عَام بُني برج خليفة؟	أين يوجد برج خليفة؟	0
مَا طريقة تحضير عجينة البيتزَا بَالحليب ؟	من طرق تحضير عجينة البيتزَا ؟	0
مَا معنَى الجهَاد؟	مَا أنوَاع الجهَاد؟	0
لمَاذَا ميدَان بيكَاديلي يجذب الكثير من السيَاح؟	مَا اسم أهم معلم سيَاحي في بريطَانيَا؟	0
كم يبلغ طول تمثَال المسيح الفَادي؟	مَا هو طول التمثَال الفَادي؟	1
إِلَى كم يصل ارتفَاع أبو الهول الموجود في مصر؟	كم يبلغ عدّ سكَان مصر؟	0
من هو المدير العَام ؟	مَا هو تعريف المدير العَام ؟	1
مَا هي ادَارة الَاعمَال ؟	مَا هي مجَالَات ادَارة الَاعمَال ؟	0
مَاهو الكوليسترول؟	مَا تعريف الكوليسترول؟	1
مَا هي أهمية الَاستثمَار ؟	الَى مَاذَا يهدف الَاستثمَار ؟	1

4 Participants and Systems

The shared task was managed using a Kaggle competition platform[3] for registration and results submissions. We have published a baseline[4] that the participants can reproduce on the same dataset.

A total of 9 teams participated in this task, with total submissions of 547, and an average of more than 60 submissions per team. In this section, we report the methodologies used for four different teams.

4.1 The Inception

The Inception team members applied different deep learning approaches, including BERT model (Devlin et al., 2018). They fine-tuned the multilingual BERT model (Devlin et al., 2018) on the sentence similarity task.

They tried various combinations of hyperparameters. For the set of parameters that made the best predictions, they repeated the experiment with different random seeds, then created an ensemble model by voting between the prediction results of these experiments. The ensemble that is composed of 3 models performed better on the public dataset while 4, 5, and 6 models have better scores on the private dataset.

4.2 Tha3aroon

Tha3aroon team did heavy work on the dataset level before building the model. First, they made sure that punctuation marks are separated from the words by making sure that characters surrounding the punctuation marks are spaces. Next, they augmented the dataset 4 different methods:

- **Positive Transitive:** If A is similar to B, and B is similar to C, then A is similar to C.

[3] https://www.kaggle.com/c/nsurl-2019-task8

[4] https://github.com/mawdoo3/q2q_workshop

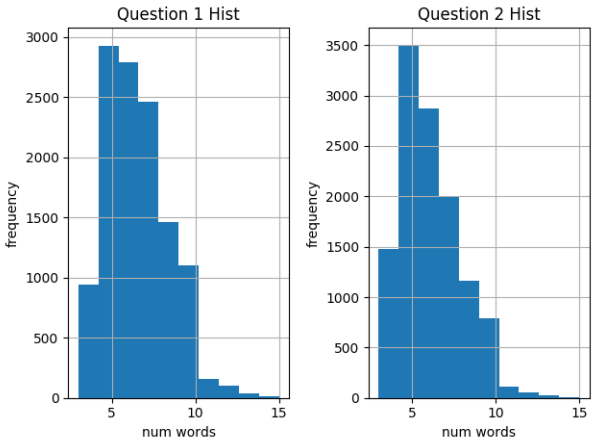


Figure 1: Distribution of question lengths (word count) in Mawdoo3 Q2Q dataset. The figure on the left shows Question 1 histogram, and Question 2 on the right.

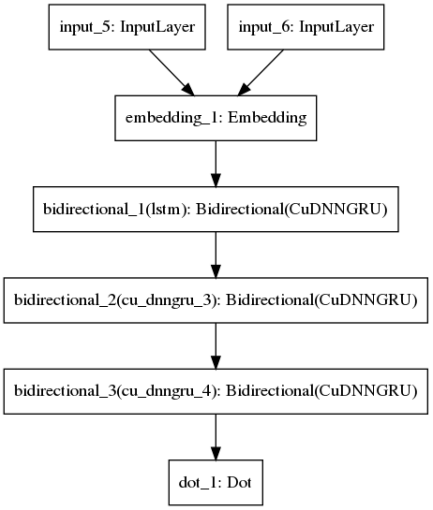


Figure 2: onekaggler model

- **Negative Transitive:** If A is similar to B, and B is NOT similar to C, then A is NOT similar to C. This rule combined with the previous one generates 5,490 extra examples (17,487 total).
- **Symmetric:** If A is similar to B then B is similar to A, and if A is not similar to B then B is not similar to A. This rule doubles the number of examples to 34,974 in total.
- **Reflexive:** By definition, a question A is similar to itself. This rule generates 10,540 extra positive examples (45,514 total) which help balance the positive and negative examples.

After the augmentation process, the training data contains 45,514 examples (23,082 positive examples and 22,432 negative ones).

To build meaningful representations for the input sequences, they used Arabic ELMo (Peters et al., 2018) pre-trained model [5] to extract contextual words embeddings and feed them as an input to the model. The model then consists of three components:

1. **Sequence representation extractor:** which takes the ELMo embeddings related to each word in the question as an input and feeds them to two special kinds of LSTM layers called Ordered Neurons LSTM (ON-LSTM) (Shen et al., 2018) and applies sequence weighted attention (Felbo et al., 2017) on the outputs of the second ON-LSTM layer to get the final question representation, this component uses the same weights to compute representations for pair questions.
2. **Merging layer:** After extracting the representations related to each question, they merged the representations using a pairwise squared distance function applied on the pair questions representation vectors.
3. **Deep neural network:** Consisting of four fully-connected layers that take the merged representation vector as an input and predicts the label using a sigmoid function as an output.

[5] https://github.com/HIT-SCIR/ELMoForManyLangs

4.3 onekaggler

The onekaggler team has built a neural network model illustrated in Figure 2. The model consists of two input layers for question1 and question2, a shared trainable word embedding layer, using Word2Vec model (Mikolov et al., 2013), initialized with Aravec tweets_cbow_300 embedding model (Soliman et al., 2017), and a stack of 3 bidirectional GRU layers with 256, 128, 64 hidden nodes, respectively. The output layer is the dot product (which calculates cosine similarity) between the outputs of the last layer of question1 and question2. The team uses mean-squared-error as a loss function alongside with Nesterov Adam optimizer. They achieve 99% accuracy on the validation set and under 94% on the test set.

4.4 Speech Translation

The Speech Translation team members have gathered feature set using sklearn's Vector-

izer Analyzer with three setups; `word-level`, `char-level`, and `char_Wb-level`. They have examined the use of n-grams (1, 2, 3, 4, and 5) for the three setups. As a preprocessing step, they applied punctuation removal, stop words filter, and text normalization. These features, combined with word stemming and POS tagging, are used for model training and testing. The team has compared the performance of a set of classifiers: BNB, LogReg, LSVM, MNB, PassAgg, PRP and SGD as well as CNN. The best performance is achieved by LSVM classifier.

5 Results and Discussion

Table 3 shows a summary of results for the participating teams. The Inception team has topped the list by achieving an accuracy score of 0.9592 using BERT models. ELMo model built by Tha3aroon scored second with an accuracy of 0.9485. This model was trained using the augmented dataset of 45,514 data samples. onekaggler team has scored third among all participants with 0.9481 accuracy using a stack of three Bidirectional GRUs. Speech Translation team has used 1 to 5 n-grams of words and characters and has experimented with several classifiers to score 0.8270, achieving the 7^{th}.

Table 3: Results for Semantic Question Similarity in Arabic. The table shows the 9 teams who participated in the workshop sorted in descending accuracy score.

#	Team Name	Score
1	The Inception	0.95924
2	Tha3aroon	0.94848
3	onekaggler	0.94809
4	Ayat Abedalla	0.91311
5	Dan Ofer	0.89465
6	heza	0.85736
7	Speech Translation	0.82698
8	AtyNegar	0.82583
9	Eyad Sibai	0.71434

One of the main takeaways is that BERT model accuracy is higher than ELMo model even when it was fine-tuned on an augmented dataset. The BERT model learns the representation of subwords while ELMo is character based model that uses convolution layers to learn word embeddings that handle out of vocabulary words. The reported results show that BERT is able to strike a good balance between a character based and word based representations and capture the word semantics for the problem of Arabic Q2Q.

Both of ELMo and BERT were able to outperform the traditional Word2Vec embeddings that is not able to capture contextual semantics nor learns subword embeddings. This proves that Arabic language (a morphologically rich language) complicates the training phase for such models because it needs to learn a completely new embedding for each morphology and is unable to generalize learnings across word variations. A word root in the Arabic language can have up to 1000 variation, Word2Vec needs to learn a number of weights equal to the number of variations multiplied by the vector size, while BERT and ELMo will only need to learn the word prefixes, roots, and word prefixes.

An interesting experiment would be to train BERT on the augmented data developed by Tha3aroon.

6 Conclusion

In this paper, we described the Arabic question similarity (Q2Q) shared the task that was organized in the workshop on NLP Solutions for Under Resourced Languages (NSURL 2019). The dataset of the shared task was made publicly available as a benchmark of this NLP task. A total of 9 teams participated in the task in which we provided a brief description of 4 of them who submitted their system description. The use of recent approaches in text embedding, i.e., BERT and ElMo, was a big factor in obtaining the best performing results. Another approach was using data augmentation that boosted up the performance. Also, an approach of using a neural network with Adam optimizer and an input layer that is initialized with pre-trained word vectors of the question pair was a well-performing solution. The ample number of participants in this workshop is an indication of the importance and interest in the Arabic language and Arabic semantic textual similarity. As future work, we would like to consider extending the task to news headlines as well as article titles.

7 Acknowledgement

We would like to thank Mawdoo3 AI data annotation team members who contributed to build and release Mawdoo3 Q2Q Dataset: Riham Badawi, Lana AlZaatreh, Raed AlRfouh, and Dana Barouqa. We would also like to thank Maw-

doo3[6] for making the datasets created for this task publicly available to support further research on Arabic Q2Q.

References

Hesham Al-Bataineh, Wael Farhan, Ahmad Mustafa, Haitham Seelawi, and Hussein T Al-Natsheh. 2019. Deep contextualized pairwise semantic similarity for arabic language questions. *arXiv preprint arXiv:1909.09490*.

Marwah Alian and Arafat Awajan. 2018. Arabic semantic similarity approaches–review. In *The 19th International Arab Conference on Information Technology (ACIT' 2018)*.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *CoRR*, abs/1803.11175.

Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity - multilingual and cross-lingual focused evaluation. *CoRR*, abs/1708.00055.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *arXiv preprint arXiv:1708.00524*.

Hua He and Jimmy Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 937–948.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3294–3302.

Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *CoRR*, abs/1703.03130.

[6] ai.mawdoo3.com

Goutam Majumder, Partha Pakray, Alexander F. Gelbukh, and David Pinto. 2016. Semantic textual similarity methods, tools, and applications: A survey. *Computación y Sistemas*, 20(4).

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119.

George A. Miller and Christiane Fellbaum. 2007. Wordnet then and now. *Language Resources and Evaluation*, 41(2):209–214.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 2786–2792.

Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016. SemEval-2016 task 3: Community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California. Association for Computational Linguistics.

Paul Neculoiu, Maarten Versteegh, and Mihai Rotaru. 2016. Learning text similarity with siamese recurrent networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP, Rep4NLP@ACL 2016, Berlin, Germany, August 11, 2016*, pages 148–157.

Chris Olah and Shan Carter. 2016. Attention and augmented recurrent neural networks. *Distill*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Mohammad Taher Pilehvar, David Jurgens, and Roberto Navigli. 2013. Align, disambiguate and walk: A unified approach for measuring semantic similarity. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 1341–1351.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2018. Coqa: A conversational question answering challenge. *CoRR*, abs/1808.07042.

Frane Saric, Goran Glavas, Mladen Karan, Jan Snajder, and Bojana Dalbelo Basic. 2012. Takelab: Systems for measuring semantic text similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2012, Montréal, Canada, June 7-8, 2012*, pages 441–448.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2018. Ordered neurons: Integrating tree structures into recurrent neural networks. *arXiv preprint arXiv:1810.09536*.

Gizem Soganciogl u, Hakime Öztürk, and Arzucan Özgür. 2017. BIOSSES: a semantic sentence similarity estimation system for the biomedical domain. *Bioinformatics*, 33(14):i49–i58.

Abu Bakr Soliman, Kareem Eissa, and Samhaa R. El-Beltagy. 2017. Aravec: A set of arabic word embedding models for use in arabic nlp. *Procedia Computer Science*, 117:256 – 265. Arabic Computational Linguistics.

Radu Soricut and Eric Brill. 2004. Automatic question answering: Beyond the factoid. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2004, Boston, Massachusetts, USA, May 2-7, 2004*, pages 57–64.

John Wieting and Douwe Kiela. 2019. No training required: Exploring random encoders for sentence classification. *CoRR*, abs/1901.10444.

Yinfei Yang, Steve Yuan, Daniel Cer, Sheng-yi Kong, Noah Constant, Petr Pilar, Heming Ge, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Learning semantic textual similarity from conversations. In *Proceedings of The Third Workshop on Representation Learning for NLP, Rep4NLP@ACL 2018, Melbourne, Australia, July 20, 2018*, pages 164–174.

NSURL-2019 Task 7: Named Entity Recognition (NER) in Farsi

Nasrin Taghizadeh, Zeinab Borhanifard, Melika Golestani Pour
School of Electrical and Computer Engineering, College of Engineering,
University of Tehran, Tehran, Iran
{nsr.taghizadeh,borhanifardz,melika.golestani}@ut.ac.ir

Mojgan Farhoodi, Maryam Mahmoudi, Masoumeh Azimzadeh
ICT Research Center, IT Faculty, Tehran, Iran
{farhoodi,mamoudy,azim-ma}@itrc.ac.ir

Heshaam Faili
School of Electrical and Computer Engineering, College of Engineering,
University of Tehran, Tehran, Iran
hfaili@ut.ac.ir

Abstract

NSURL-2019 Task 7 focuses on Named Entity Recognition (NER) in Farsi. This task was chosen to compare different approaches to find phrases that specify Named Entities in Farsi texts, and to establish a standard testbed for future researches on this task in Farsi. This paper describes the process of making training and test data, the list of participating teams (6 teams), and evaluation results of their systems. The best system obtained 85.4% of F_1 score based on phrase-level evaluation on seven classes of NEs including person, organization, location, date, time, money, and percent.

1 Introduction

Named Entity Recognition (NER) is defined as the task of identifying relevant nouns such as persons, products, and genes which are mentioned in a text. NER is an important task as it is usually employed as a primary step in the other tasks such as event detection from news, customer support for on-line shopping, knowledge graph construction, and biological analysis (Bokharaeian et al., 2017).

NER is a famous and well-studied task in English (Yadav and Bethard, 2018) and some other languages like Arabic (Shaalan, 2014; Helwe and Elbassuoni, 2019; Taghizadeh et al., 2018) and German (Riedl and Padó, 2018). However, this task is not highly examined in Farsi because there is no standard benchmark for it. Although there are some Farsi NER corpora such as PEYMA (Shahshahani et al., 2018), ArmanPersoNER (Poostchi et al., 2016), A'laam(Hosseinnejad et al., 2017), and Persian-NER[1]; none of them is known as standard data set to the research community. Moreover, the type of named entities and annotation guidelines are different in each corpus. Because of the diversity of annotation types and data sets which were used for training and test, the result of current researches on Farsi NER cannot be directly compared.

The goal of this competition was to bring Farsi NER researchers together. We introduce a large scale corpus containing about 900K tokens as the training data for this task. To evaluate the participating teams, a test set was prepared which contains 150K tokens. The training and test set follow the same annotation schema. These data sets are publicly available for further researches[2]. The domain of all data is the news sentences because they are the most entity-rich.

[1] https://github.com/Text-Mining/Persian-NER
[2] https://github.com/nasrin-taghizadeh/NSURL-Persian-NER

Participants were allowed to use any public data and resources such as Farsi Wikipedia[3] and Farsi Knowledge Graph[4] (Sajadi et al., 2018) in addition to the official training data of the shared task in the process of making their system. In this case, they must thoroughly describe those resources and the way they used them.

To the best of our knowledge, this is the first shared task in Farsi. Since Farsi belongs to the group of low-resource languages (Taghizadeh and Faili, 2016; Fadaei and Faili, 2019), the availability of annotated corpora and resources will be very useful for future investigation in this language.

2 Farsi NER

So far, some researchers have been conducted on Farsi NER. Poostchi et al. (Poostchi et al., 2018) presented a BiLSTM-CRF model, which is a recurrent neural network obtained by a combination of a long short-term memory (LSTM) and a conditional random field (CRF). They presented a public data set for Farsi NER, called ArmanPersoNER, which includes six types of NEs: person, organization, location, facility, product, and event. Their model showed 77.45% of F_1 on ArmanPersoNER.

Shahshahani et al. (Shahshahani et al., 2018) presented a hybrid system consisting of a rule-based and a statistical system. The rule-based system composed of a large list of NEs in Farsi in addition to the regular expressions for detecting them. The statistical system is a CRF model trained by the PEYMA corpus. Their system reached 84% of F_1 for seven classes of person, organization, location, date, time, money, and percent, based on 5-fold validation on the training data.

Hossinnejad et al. (Hosseinnejad et al., 2017) presented a corpus named A'laam consisting of 13 classes of named entities. They split this corpus into two parts of 90% and 10% for the training and test, respectively, and trained a CRF model using the training part. They obtained 92.9% and 78.5% of precision and recall, respectively.

Zafarian et al. (Zafarian et al., 2015) proposed a semi-supervised method for Farsi NER. They used an un-labeled bilingual data in addition to a small labeled data to train their system. They presented a bootstrap method that iteratively trains a CRF model using the labeled data as well as those unlabeled data that the current model predicts them with high confidence. Their data contains three classes of person, organization, and location. They reached 67.5% of F_1.

Current researches on Farsi NER use different data for the training and test. Most of these data are not public or annotated with diverse annotation schema. The evaluation methods of them are not similar and so their results cannot be directly compared.

3 The Task

Participating systems have to predict NE tags for a set of tokenized documents. We defined two subtasks:

- 3-classes including person, organization, and location;
- 7-classes including date, time, money, and percent in addition to the three above classes.

NEs that belong to four classes of date, time, money, and percent sometimes can be recognized using the rule-based or hybrid methods (Ahmadi and Moradi, 2015; Riaz, 2010); while NEs of the classes of person, organization, or location are often recognized based on the gazetteer lists and they are more subject to ambiguity. Therefore, we have separated these two subtasks and participants could submit different systems for them.

3.1 Baseline Method

CoNLL 2003 defined the baseline of NER task a system which only selects complete unambiguous named entities that appear in the training data. We adapted this baseline as follows:

- In case of overlap between two candidate named entities, the longer is selected. For example, consider three NEs of the training data: 1) "ایران/Iran" which is a *location*, 2) "مجلس شورای اسلامی/Islamic Consultative Assembly" which is an *organization*, and 3) "مجلس شورای اسلامی ایران/Islamic

[3] https://fa.wikipedia.org/wiki/
[4] http://farsbase.net/search/html/index.html

Consultative Assembly of Iran" which is an *organization* as well. To extract named entities from phrase "مجلس شورای اسلامی ایران/Islamic Consultative Assembly of Iran", the baseline system selects whole phrase as an *organization* instead of separately tagging "مجلس شورای اسلامی/Islamic Consultative Assembly" as an *organization* and "ایران/Iran" as a *location*.

- When two NEs are next to each other and have the same tag, they are merged. For example, there are two NEs in the training data: "۲۲ بهمن/22th of Bahman" and "۱۳۵۷/1357" which are *date*. In the test phase, the baseline system visits phrase "۲۲ بهمن ۱۳۵۷/22th of Bahman 1357", and separately selects these two phrases as *date*. Then, they are merged to be one mention. Our analysis showed that this heuristic is often true. However, in few cases it may be wrong. For example, consider following sentence:

  گزینه اول جانشینی آمانو کورنل فروتا دیپلمات رومانیایی است.

  "The first option for Amano's successor is Cornel Fruta, a Romanian diplomat."

  There are two adjacent mentions with the same *person* type: "آمانو/Amano" and "کورنل فروتا/Cornel Feruta", and merging them into one NE is not correct.

These examples reveal some challenges of Farsi NER. One challenge is that a unique named entity may appear in the text with different names. For example, "مجلس شورای اسلامی ایران/Islamic Consultative Assembly of Iran", "مجلس شورای اسلامی/Islamic Consultative Assembly" and "مجلس/assembly" are different names of the same entity. While "مجلس/assembly" is a common noun, it names an *organization*. It means that gazetteers are not sufficient for detecting boundaries of entity mentions.

Another challenge is that two or more entity mentions may be adjacent in the sentence, in the sense that there is no word between them. They may have different or similar types. In case of similar types, it may be possible or not to merge them into a unique mention. For example adjacent entity mentions of *date*, mostly can be merged, such as "بیست و هفتم مهر ماه سال 1398".

4 Data Set Creation

We presented a training data set which has two parts: the first part is PEYMA corpus (Shahshahani et al., 2018) containing 300K tokens; the second part has 600K tokens. The same annotation schema was used for annotating two parts. This annotation schema was prepared based on two standard guidelines: 1) MUC[5] and 2) CoNLL[6]; then it was adapted for Farsi linguistic structures (Shahshahani et al., 2018). In these data sets there are seven classes of named entities: *person*, *organization*, *location*, *money*, *date*, *time*, and *percent*.

Steps of creating data set include news collection, pre-processing, and named entity tagging. The test data has two parts: in-domain and out-of-domain. The former was sampled from the same news websites in the same period of time that the training data were collected. The latter was selected from different news websites at different times. Specifically, documents of the training data mostly were sampled from a few Farsi news websites between 2016 and 2017; while out-of-domain documents were sampled from multiple Farsi news websites from different countries of the world mainly in 2019. Therefore, in-domain documents are more similar in word distribution to the training data than the out-of-domain documents.

Therefore, in-domain documents are more similar in word distribution to the training data than the out-of-domain documents. Preprocessing on news documents was performed using Persianp toolkit (Mohseni et al., 2016), which includes tokenization, sentence split, and normalization. Two annotators performed the annotation task, and the agreement between them is 95% which shows the quality of the annotations. The data format is similar to the CoNLL 2003, in which each line contains one word and empty lines represent sentence boundaries. Annotation format is IOB that encodes the beginning and inside of the entity mentions and type of them.

[5] https://www-nlpir.nist.gov/related_projects/muc/proceedings/ne_task.html

[6] https://www.clips.uantwerpen.be/conll2003/ner/

Table 1: Data Statistics

	Lang	#Doc	#Sent	#Tokens
Training Data	Fa	1,456	27,130	885,296
Test Data	Fa	431	4,154	144,526
ArmanPersoNER	Fa	-	7,682	250,015
CoNLL-2003	En	1,393	22,137	301,418
CoNLL-2003	Gr	909	18,973	310,318

Table 2: Statistics of Test data

Test Data	#Doc	#Sent	#Tokens
In-domain	196	1,571	68,063 (47%)
Out-of-domain	235	2,583	76,463 (53%)

4.1 Data Statistics

Table 1 represents general statistics of our Farsi data sets including the number of articles, sentences, and tokens in comparison with English and German data sets of the CoNLL 2003. The comparison reveals that the Farsi training data is a large scale data set that can be used for further researches on Farsi NER. Table 2 shows details of the in-domain and out-of-domain parts of the test data. The two parts have a nearly equal number of tokens. Tables 3 and 4 represent the total number of phrases and the number of unique phrases tagged for each class of named entities in the training and test data. Considering the size of each corpus, the test set is denser in terms of the entity tags.

5 Participating Systems and Results

Six teams have participated in both subtasks. Most of them opted for use of CRF models and deep learning methods specifically Bi-LSTM. Because these two models deal with sequence tagging problems. Word embeddings, n-grams, and POS tags were used as features by the systems. Morphological and orthographic features of Farsi phrases were used by some of the participants. Table 5 briefly shows the models and features used by the participants.

Table 3: Number of total phrases tagged per class

Data	PER	ORG	LOC	MON	DAT	TIM	PCT	Total
Training	12,495	14,205	15,403	1,294	4,467	571	997	49,432
Test	2,738	3,160	4,081	357	1,147	165	156	11,804

Table 4: Number of unique phrases tagged per class

Data	PER	ORG	LOC	MON	DAT	TIM	PCT	Total
Training	5,228	4,547	2,738	1,008	1,910	338	453	16,020
Test	1,470	1,326	1,015	288	628	114	97	4,917

5.1 Evaluation Metrics

There are different methods for the evaluation of NER systems. Two main methods are phrase-level and word-level evaluation. In the phrase-level evaluation, a phrase is counted as true-positive for class c, if both boundaries of the phrase and its predicted tag are correct. In contrast, in word-level evaluation, each word is considered separately. Therefore, the phrase-level evaluation is tougher than the word-level evaluation.

We used evaluation script of conlleval[7]. This script computes three measures including precision, recall, and F_1 based on the standard definition. Evaluation of the 3-classes subtask has been performed based on the macro-averaging method. Accordingly, precision and recall are obtained by averaging of the precision and recall of the three classes of *person*, *organization*, and *location*.

Evaluation of 7-classes subtask has been conducted using the micro-averaging method due to class imbalance problem, in the sense that frequencies of NE phrases belonging to four classes of *date*, *time*, *money*, and *percent* are very fewer than the three classes of *person*, *organization* and *location*, according to the Tables 3 and 4. So, in this case, the micro-averaging better evaluates the quality of systems.

5.2 Result

Participating teams mainly used sequence tagging methods including CRF and Bi-LSTM networks. The feature sets used by them include lexical, morphological, and structural features. Tables 6 and 7 show the evaluation results of 3-classes and 7-classes subtasks, respectively. Generally, results of the word-level evaluation are higher than the phrase-level evaluation. Moreover, the results of the evaluation by the in-domain data are higher than the out-of-domain data in terms of the F_1 score. All teams outperformed the baseline

[7] https://github.com/sighsmile/conlleval

Table 5: Description of Participating Systems

Team	Model	Word Embeddings	Features
MorphoBERT	BERT + BiLSTM	BERT for token representation word2vec for word clustering	cluster number of words, morphology
Beheshti-NER-1	Transformer-CRF	BERT	-
Team-3	CRF	-	-
ICTRC-NLPGroup	CRF	-	n-gram, lemma, linguistics rules
UT-NLP-IR	CRF	-	POS, NP-chunk, word n-gram, char n-gram, stem, lemma
SpeechTrans	SVM	-	word unigram, char 5-grams, POS, stem, normalized surface
Baseline	heuristic	-	-

and ranking of the teams are the same based on all kinds of the evaluations.

The best F_1 scores are 85.9% and 88.5% based on the phrase-level and word-level evaluation, respectively, which are obtained by the MorphBERT system (Mohseni and Tebbifakhr, 2019). The second best system, Beheshti-NER-1 (Taher et al., 2019), got near F_1 scores: 84.0% and 87.9% based on the phrase-level and word-level evaluation, respectively. These two systems used BERT model (Devlin et al., 2018) for training high accurate representation of Farsi tokens. BERT is a deep bi-directional language model that presented state-of-the-art results in a wide variety of NLP tasks. Both systems used the BERT to process a huge amount of un-labeled Farsi texts to obtain pre-trained word embeddings which then was fine-tuned for the NER task.

MorphoBERT used a morphological analyzer as a prior step before the BERT network. Farsi is rather rich-morphology and analyzing tokens to find their parts reveals the grammatical and semantic information. So, instead of embedding tokens of sentences into the network, MorphoBERT firstly decomposes tokens into constituents and then fed these constituents into the BERT network. Then, the representation of the sentence which was obtained from the BERT is given to a Bi-LSTM network. Additionally, a vector representing word cluster features is given to the Bi-LSTM. Finally, the Softmax layer produces a probability distribution over all classes (Mohseni and Tebbifakhr, 2019).

Beheshti-NER-1 system utilizes a CRF model on top of the BERT network. The motivation of using CRF is that an encoder like BERT tries to maximize the likelihood by selecting best-hidden representations, and CRF tries to maximize the likelihood by selecting best output tags (Taher et al., 2019).

To better understand the details of the scores, we presented the F_1 scores of each 7 classes based on the phrase-level evaluation in Table 8. Generally, the most F_1 scores were obtained by *percent* and *money* classes. Because there are specific keywords representing them and so there are high-precision patterns that specify entity mentions of these classes. Specifically, *percent* often comes with the keywords like "درصد/percent"; while *money* appears with words and phrases denoting money like "دلار/Dollar", "ریال/Rial", or "یورو/Euro". On the other hand, the least F_1 scores were obtained by the *time* class. Perhaps because the number of phrases in the training data having *time* tag is very few in comparison to the other classes.

6 Conclusion

We have described the NSURL-2019 task 7: NER in Farsi. Six systems have processed the Farsi NE data. The best performance was obtained by the MorphoBERT system that is 85.4% of F_1 score based on the phrase-level evaluation of the 7-classes subtask. This system uses morphological features of Farsi words together with the BERT model and Bi-LSTM.

References

Farid Ahmadi and Hamed Moradi. 2015. A hybrid method for Persian Named Entity Recognition. In *2015 7th Conference on Information and Knowledge Technology (IKT)*, pages 1–7. IEEE.

Table 6: Evaluation of systems for subtask 3-classes

	Team	Test Data																	
		Phrase-level evaluation									Word-level evaluation								
		In-domain			*Out-of-domain*			*Total*			*In-domain*			*Out-of-domain*			*Total*		
		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
1	MorphoBERT	88.7	85.5	87.1	86.3	83.8	85.0	87.3	84.5	85.9	92.5	86.7	89.5	91.5	84.0	87.6	92.1	85.2	88.5
2	Beheshti-NER-1	85.3	84.4	84.8	84.4	82.6	83.5	84.8	83.3	84.0	90.5	87.2	88.8	89.7	85.0	87.3	90.1	85.8	87.9
3	Team-3	87.4	77.2	82.0	87.4	73.4	79.8	87.4	75.0	80.7	89.2	79.5	84.1	89.5	74.7	81.4	89.3	76.9	82.7
4	ICTRC-NLPGroup	87.5	76.0	81.3	86.2	69.6	77.0	86.8	72.3	78.9	90.1	78.2	83.7	88.7	70.2	78.4	89.4	73.5	80.7
5	UT-NLP-IR	75.3	68.9	72.0	72.3	60.7	66.0	73.6	64.1	68.5	87.3	71.9	78.9	86.4	61.1	71.6	86.9	65.7	74.8
6	SpeechTrans	41.5	39.5	40.5	43.1	38.7	40.8	42.4	39.0	40.6	66.8	38.3	48.7	66.2	35.2	46.0	66.6	36.4	47.0
7	Baseline	32.2	45.8	37.8	32.8	39.1	35.7	32.5	41.9	36.6	46.2	42.6	44.3	45.2	35.1	39.5	45.9	38.4	41.8

Table 7: Evaluation of systems for subtask 7-classes

	Team	Test Data																	
		Phrase-level evaluation									Word-level evaluation								
		In-domain			*Out-of-domain*			*Total*			*In-domain*			*Out-of-domain*			*Total*		
		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
1	MorphoBERT	88.4	84.8	86.6	86.0	83.1	84.5	87.0	83.8	85.4	94.0	89.1	91.5	91.8	85.7	88.6	92.8	87.1	89.9
2	Beheshti-NER-1	84.8	83.6	84.2	83.9	82.0	83.0	84.3	82.7	83.5	91.4	87.3	89.3	89.7	85.7	87.7	90.4	86.5	88.4
3	Team-3	87.4	77.3	82.0	87.3	72.8	79.4	87.3	74.7	80.5	91.3	84.1	87.5	90.9	77.9	83.9	91.1	80.7	85.5
4	ICTRC-NLPGroup	87.0	76.1	81.2	86.2	70.2	77.4	86.5	72.7	79.0	89.2	83.1	86.1	89.8	76.5	82.6	89.7	79.4	84.2
5	UT-NLP-IR	77.3	70.2	73.6	74.1	61.9	67.5	75.5	65.4	70.1	92.7	79.3	85.4	91.1	68.4	78.1	91.9	73.1	81.4
6	SpeechTrans	38.0	34.5	36.2	38.9	33.6	36.0	38.5	34.0	36.1	76.1	32.9	45.9	74.9	30.3	43.2	75.7	31.5	44.5
7	Baseline	32.8	45.7	38.2	32.0	38.1	34.8	32.4	41.3	36.3	50.6	47.8	49.2	42.6	35.1	38.5	46.5	40.9	43.5

Table 8: Details of phrase-level evaluation for subtask 7-classes (values are F_1 score)

	Team	Named Entity Classes							F_1
		PER	ORG	LOC	DAT	TIM	MON	PCT	
1	MorphoBERT	90.4	80.3	87.1	78.9	71.0	93.6	96.8	85.4
2	Beheshti-NER-1	81.8	80.8	88.0	77.8	75.8	85.1	91.6	83.5
3	Team-3	79.9	77.2	83.9	74.7	64.3	92.1	97.4	80.5
4	ICTRC-NLPGroup	76.2	75.93	82.8	76.0	67.1	91.3	93.6	79.0
5	UT-NLP-IR	63.4	58.8	78.2	76.1	69.1	84.5	93.5	70.1
6	SpeechTrans	24.3	23.5	63.1	12.0	4.1	0.3	0.7	36.1
7	Baseline	23.5	38.1	44.2	41.6	30.3	13.7	36.6	36.3

Behrouz Bokharaeian, Alberto Diaz, Nasrin Taghizadeh, Hamidreza Chitsaz, and Ramyar Chavoshinejad. 2017. SNPPhenA: a corpus for extracting ranked associations of single-nucleotide polymorphisms and phenotypes from literature. *Journal of biomedical semantics*, 8(1):14.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Hakimeh Fadaei and Heshaam Faili. 2019. Using syntax for improving phrase-based SMT in low-resource languages. *Digital Scholarship in the Humanities*.

Chadi Helwe and Shady Elbassuoni. 2019. Arabic Named Entity Recognition via deep co-learning. *Artificial Intelligence Review*, 52(1):197–215.

Shadi Hosseinnejad, Yasser Shekofteh, and Tahereh and Emami Azadi. 2017. A'laam Corpus: A Standard Corpus of Named Entity for Persian Language. *Signal and Data Processing*, 14(3).

Mahdi Mohseni, Javad Ghofrani, and Heshaam Faili. 2016. Persianp: a Persian text processing toolbox. In *Computational Linguistics and Intelligent Text Processing - 17th International Conference, CICLing 2016, Konya, Turkey, April 3-9, 2016, Revised Selected Papers, Part I*, pages 75–87.

Mahdi Mohseni and Amirhossein Tebbifakhr. 2019. MorphoBERT: a Persian NER system with BERT and morphological analysis. In *Proceedings of the first International Workshop on NLP Solutions for Under Resourced Languages*, NSURL '19, Trento, Italy.

Hanieh Poostchi, Ehsan Zare Borzeshi, Mohammad Abdous, and Massimo Piccardi. 2016. PersoNER: Persian Named Entity Recognition. In *COLING 2016-26th International Conference on Computational Linguistics, Proceedings of COLING 2016: Technical Papers*.

Hanieh Poostchi, Ehsan Zare Borzeshi, and Massimo Piccardi. 2018. BiLSTM-CRF for Persian Named-Entity Recognition armanpersonercorpus: the first entity-annotated persian dataset. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

Kashif Riaz. 2010. Rule-based named entity recognition in Urdu. In *Proceedings of the 2010 named entities workshop*, pages 126–135. Association for Computational Linguistics.

Martin Riedl and Sebastian Padó. 2018. A Named Entity Recognition shootout for German. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 120–125.

Mohamad Bagher Sajadi, Behrouz Minaei, and Ali Hadian. 2018. Farsbase: A cross-domain farsi knowledge graph. In *SEMANTICS Posters&Demos*.

Khaled Shaalan. 2014. A survey of Arabic Named Entity Recognition and classification. *Computational Linguistics*, 40(2):469–510.

Mahsa Sadat Shahshahani, Mahdi Mohseni, Azadeh Shakery, and Heshaam Faili. 2018. PEYMA: A Tagged Corpus for Persian Named Entities. *ArXiv*, abs/1801.09936.

Nasrin Taghizadeh, Heshaam Faili, and Jalal Maleki. 2018. Cross-Language Learning for Arabic Relation Extraction. *Procedia computer science*, 142:190–197.

Nasrin Taghizadeh and Hesham Faili. 2016. Automatic Wordnet Development for Low-Resource Languages using Cross-lingual WSD. *Journal of Artificial Intelligence Research*, 56:61–87.

Ehsan Taher, Seyed Abbas Hoseini, and Mehrnoush Shamsfard. 2019. Beheshti-NER: Persian named entity recognition using BERT. In *Proceedings of the first International Workshop on NLP Solutions for Under Resourced Languages*, NSURL '19, Trento, Italy.

Vikas Yadav and Steven Bethard. 2018. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158.

Atefeh Zafarian, Ali Rokni, Shahram Khadivi, and Sonia Ghiasifard. 2015. Semi-supervised learning for named entity recognition using weakly labeled training data. In *2015 The International Symposium on Artificial Intelligence and Signal Processing (AISP)*, pages 129–135. IEEE.

Yorùbá Gender Recognition from Speech using Attention-based BiLSTM

Ibukunola A. Modupe
Department of ICT
Vaal University of Technology
ibukunolam@vut.ac.za

Tshephisho J. Sefara
Next Generation Enterprises and Institutions
Council for Scientific and Industrial Research
tsefara@csir.co.za

Sunday O. Ojo
Department of Computer Science
Tshwane University of Technology
ojoso@tut.ac.za

Abstract

Gender recognition in speech processing is one of the most challenging tasks. While many studies rely on extracting features and designing enhancement classifiers, classification accuracy is still not satisfactory. The remarkable improvement in performance achieved through the use of neural networks for automatic speech recognition has encouraged the use of deep neural networks in other voice techniques such as speech, emotion, language and gender recognition. An earlier study showed a significant improvement in the gender recognition of pictures and videos. In this paper, speech is used to create a gender recognition scheme based on neural networks. Attention-based BiLSTM architecture is proposed to discover the best approach for gender identification in Yorùbá. Acoustic features, including time, frequency, and cepstral features are extracted to train the model. The model obtained the state-of-the-art performance in speech-based gender recognition with 99% accuracy and F_1 score.

1 Introduction

Gender recognition is an important topic in signal processing and can be applied in mobile healthcare system (Alhussein et al., 2016), facial recognition (Hwang et al., 2009), and age classification (Chen et al., 2011). Applications of gender recognition system includes tasks such as (Mukherjee and Liu, 2010): (i) Verifying a customer when making telephone bank transaction, (ii) Security measure when retrieving confidential information, (iii) Forensic, (iv) Surveillance, (v) and Blog authorship. Recognition of gender from the speech is a challenging task with these increasing number of systems in real-life. Recent hardware and software development allowed new techniques and methods to be explored to improve the efficiency of gender recognition systems. Gender classification systems from speech signal are affected by the performance of the recording tools, the language of the speaker, and noisy recording settings. As a result, to obtain adequate classification results, gender recognition from speech signals requires valid classifiers and feature extractors. In the areas of machine learning and computer vision, deep neural networks (DNNs) have shown notable achievements (Moghaddam and Ming-Hsuan Yang, 2000; Hwang et al., 2009). Deep neural networks, after thorough training, can effectively extract and classify different feature sets. DNNs are most effective when the training set contains a complicated feature space that needs high-level representation. In this paper, deep recurrent neural networks (DRNNs) are used as classifiers and gender-recognition extractors. Bidirectional long-short term memory (BiLSTM) is combined with an attention mechanism to learn the features. Because gender recognition is a binary classification, a sigmoid activation function has been used to classify the gender.

1.1 Motivation

Gender recognition systems for well-resourced languages like English are available, but for African languages like Yorùbá are not available. Yorùbá is a Niger-Congo language related to Igala, Edo, Ishan, and Igbo amongst others. It is one of the official languages of Nigeria and spoken in a couple of countries on the West African coast. An estimated 20+ million people speak Yorùbá as their first language in southwestern Nigeria and more in the Republics of Benin and Togo. Yorùbá is also spoken by diaspora communities of traders in Cote d'Ivoire, Ghana, Senegal and the Gambia, and it used to be a vibrant language in Freetown, Sierra Leone. Outside West Africa, millions of people have Yorùbá language and culture as part of their heritage; Yorùbá religion being one of the

means of survival in Cuba during the obnoxious slave trade. Many who did not have Yorùbá as their heritage bought into Yorùbá identity through religious transformation. Yorùbá language, culture and religion survived since then until now in Brazil and various other New World countries (Atanda et al., 2013; Pulleyblank et al., 2017). Yorùbá is identified as one of the under-resourced languages (Besacier et al., 2014), few systems for under-resourced African languages has been developed (Sefara et al., 2016; Sefara et al., 2019; Sefara et al., 2017; Sefara and Manamela, 2016; Sefara et al., 2016; Van Niekerk and Barnard, 2012; Modipa and Davel, 2015; Manamela et al., 2018; Mokgonyane et al., 2019). While the development of speech-based systems for Yorùbá is an open research, it is essential to continue to create a Yorùbá gender recognition system that may later help other researchers and to strengthen the cultural identify of the language.

The main contributions of this paper can be listed as below.

- A new classifier architecture is proposed. A BiLSTM architecture with attention mechanism is used.
- Acoustic features such as Time, Frequency, and Cepstral-domain features are used for gender recognition.
- We release the code[1] used in this paper.

The rest of the paper is organized as follows: Section 2 gives the literature review on gender recognition. Section 3 details the features, learning models, and evaluation methods. Section 4 discusses the experimental results, and the paper is concluded in Section 5.

2 Literature Review

Gender recognition can be approached from text (Mukherjee and Liu, 2010), images (Moghaddam and Ming-Hsuan Yang, 2000; Hwang et al., 2009; Kumar et al., 2019; Qawaqneh et al., 2017a), videos (Ding and Ma, 2011; Chen et al., 2017), accelerometers (Bales et al., 2016), wearables (Gümüşçü et al., 2018), and speech (Harb and Chen, 2003; Azghadi et al., 2007; Meena et al., 2013) to train machine learning models and neural networks for classification. Meena et al. (2013) proposed a novel gender classification technique in speech processing using neural network and fuzzy logic. Authors used acoustic features such as short time energy, zero crossing rate and energy entropy. Their work can be expanded by not only using time domain features but also to include frequency and cepstral domain features. An example of cepstral-domain features are Mel Frequency Cepstral Coefficients (MFCCs). Qawaqneh et al. (2017a) used MFCCs, fundamental frequency (F0) and the shifted delta cepstral coefficients (SDC) to train a jointly fine-tuned deep neural networks. Their model obtained accuracy of 64%. Conversely, Harb and Chen (2003) did not use MFCCs but used Mel Frequency Spectral Coefficients (MFSC) to train a gender identification system using neural networks. Authors showed that smoothing improves the accuracy of the model and MFSC features were better than MFCC features. Azghadi et al. (2007) used acoustic features and pitch features to train a gender classification system based on feed-forward back-propagation neural network. Their model obtained an accuracy of 96%. Qawaqneh et al. (2017b) introduced shared class labels among misclassified labels to regularize the DNN weights and to generate transformed MFCCs feature set using Backus-Naur Form (BNF). Authors used DNN and i-vector models to build age and gender classification system. The BNF-DNN obtained accuracy of 58.98 and BNF-I-vector obtained 56.13

Machine learning algorithm are used for gender recognition. Chaudhary and Sharma (2018) used support vector machines (SVMs) to train a gender identification system based on voice signal by extracting the features such as pitch, energy and MFCC. Their model obtained accuracy of 96.45%. Gaussian mixture models (GMMs) and multilayer perceptrons (MLPs) are used in (Djemili et al., 2012) to create a gender identification system. The models obtained accuracy of 96.4% using MFCCs as features. Jadav (2018) proposed a voice-based gender identification using machine learning. Author extracted acoustic features to train a SVM which obtained testing accuracy of 97%.

3 Methodology

The architecture of a gender recognition system is shown in Figure 1. The system consists of the training and prediction phases.

[1] https://github.com/SefaraTJ/yoruba-gender-recognition/

- In the training phase, the speech signal is inputted to the system, and pre-processing occurs (noise removal, dimensionality reduction). Acoustic features are extracted. Then a machine learning model is built and trained on the extracted features.

- In recognition phase, an unlabelled or unknown speech signal is inputted to the system. The model predicts and outputs the gender of the inputted signal.

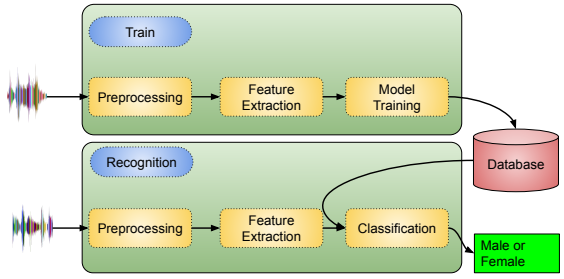


Figure 1: Architecture of a gender recognition system.

3.1 Data

We obtained speech database from (van Niekerk et al., 2015) used in (Van Niekerk and Barnard, 2012), where recordings consist of 16 female and 17 male recordings in Yorùbá. About 130 utterances were read from short texts for each speaker. The length of the recordings is 165 minutes. The audios are 16 bit PCM at 16kHz sampling rate.

We use Principal Component Analysis (PCA) (Moore, 1981; Ding and He, 2004) to explore the data in Figure 2 by scaling to 2 dimension. The centers are illustrated using k-means (Ding and He, 2004) with $k = 2$. We observe the data can be separated into males and females. This will simplify the learning of the models.

3.2 Feature Extraction

Feature extraction is the transformation of original data into a dataset that contains the most discriminatory information, with reduced numbers of variables. The 34 acoustic features shown in Figure 3 are extracted from the short-term windows with frame size of 50ms at a Hamming window of 25ms using a library in (Giannakopoulos, 2015). The final feature vector contains the mean and standard deviation which sums to feature size of 68. The features can be grouped into three categories:

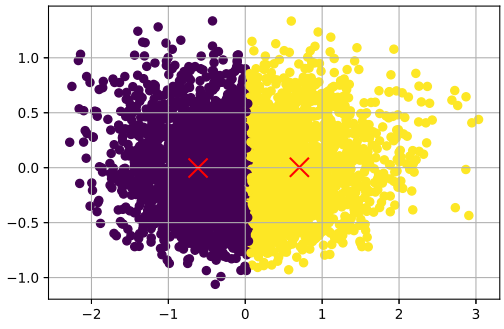


Figure 2: PCA showing gender clusters and k-means showing cluster centres.

- Time-domain features (Zero Crossing Rate, Energy, and Entropy of Energy).

- Frequency-domain features (Spectral Spread, Spectral Centroid, Spectral Flux, Spectral Entropy, Spectral Rolloff, Chroma Deviation, Chroma Vector).

- Cepstral-domain features - includes MFCCs that has an ability to model the vocal tract filter.

Feature ID	Feature Name	Description
1	Zero Crossing Rate	The rate of sign-changes of the signal during the duration of a particular frame.
2	Energy	The sum of squares of the signal values, normalized by the respective frame length.
3	Entropy of Energy	The entropy of sub-frames' normalized energies. It can be interpreted as a measure of abrupt changes.
4	Spectral Centroid	The center of gravity of the spectrum.
5	Spectral Spread	The second central moment of the spectrum.
6	Spectral Entropy	Entropy of the normalized spectral energies for a set of sub-frames.
7	Spectral Flux	The squared difference between the normalized magnitudes of the spectra of the two successive frames.
8	Spectral Rolloff	The frequency below which 90% of the magnitude distribution of the spectrum is concentrated.
9-21	MFCCs	MFCCs form a cepstral representation where the frequency bands are distributed according to the mel-scale.
22-33	Chroma Vector	A 12-element representation of the spectral energy where the bins represent the 12 equal-tempered pitch classes of western-type music (semitone spacing).
34	Chroma Deviation	The standard deviation of the 12 chroma coefficients.

Figure 3: Acoustic features (Giannakopoulos, 2015).

3.3 Feature Normalization

Is an crucial step for gender recognition using speech. The goal is to remove speaker and record-

ing variability. We normalize features by removing the mean and scaling to a unit variance using the following normalization equation. For normalized feature $\hat{y}$:

$$\hat{y} = \frac{x - \mu}{\sigma} \quad (1)$$

where σ represents the variance and μ represents the mean for each feature vector x.

3.4 The Classifier Model

This section explains the proposed BiLSTM model. As shown in Figure 4, the first layer is the input layer having the same size of the input vector. Followed by the BiLSTM layer having 128 units. Followed by the attention layer, followed by LSTM layer, followed by 4 dense layers with the last layer activated by the *sigmoid* function.

3.4.1 BiLSTM Layer

For this gender recognition problem, we model the speech signal using recurrent neural network (RNN), specifically BiLSTM. LSTM was introduced by Hochreiter and Schmidhuber (1997), has shown to be stable and accurately model long-time dependencies in different tasks like speech recognition, machine learning, and computer vision (Moghaddam and Ming-Hsuan Yang, 2000; Hwang et al., 2009). BiLSTM trains two LSTMs on the input sequence. The second LSTM is a reverse copy of the first one, the aim is to capture past and future input features for a specific time step.

3.4.2 Attention Layer

Attention is a mechanism allowing neural networks to examine specific areas of the input speech signal in more detail to decrease the task complexity and to exclude irrelevant information. An attention layer is included for determining the contribution of each signal frame to the whole speech signal. The attention mechanism assigns a weight w_i to each frame feature h_i. The hidden state is lastly calculated by a weighted sum function to generate a hidden acoustic feature vector r. Formally:

$$p_j = \tanh(W_h h_j + b_h), \quad p_j \in [-1, 1] \quad (2)$$

$$w_j = \frac{exp(p_j)}{\Sigma_{t=1}^{N} exp(p_t)}, \quad \Sigma_{j=1}^{N} w_j = 1 \quad (3)$$

$$r = \Sigma_{j=1}^{N} w_j h_j, \quad r \in R^{2L} \quad (4)$$

where W_h and b_h are the weight and bias from the attention layer.

3.4.3 Dense Layer

The attention layer is followed by four dense layers with different sizes of neurons. The output of attention layer is fed into first dense layer with 128 hidden neurons activated by *rectified linear unit*. And to avoid overfitting, we add a dropout layer having probability of 0.5 between the first three dense layers that have 128, 64, and 32 neurons respectively. The last dense layer uses *sigmoid* activation function to create binary classification. The *sigmoid* activation function is defined as follows:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (5)$$

3.5 Evaluation

This section describes the performance measurements used to evaluate model quality. The performance of the model is affected by the speech signal quality, the training data size, and most importantly the optimization of learning algorithm. The following evaluation metrics are applied:

Accuracy represents all correctly predicted samples, calculated as follows:

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \quad (6)$$

Binary cross entropy is a Sigmoid activation plus a Cross Entropy loss. We use binary cross entropy loss function since the labels of the data are binary. It is calculated as follows:

$$-(y \log(p) + (1 - y) \times \log(1 - p)) \quad (7)$$

where p is the probability predicted by the model.

Precision is the total number of the positively predicted examples that are relevant. It is calculated as follows:

$$Precision = \frac{tp}{tp + fp} \quad (8)$$

Recall measures how well a model is at predicting the positives. It is calculated as follows:

$$Recall = \frac{tp}{tp + fn} \quad (9)$$

F_1 **score** is the harmonic mean of precision and recall. It is calculated as follows:

$$F_1 score = 2 \times \frac{precision \times recall}{precision + recall} \quad (10)$$

where:

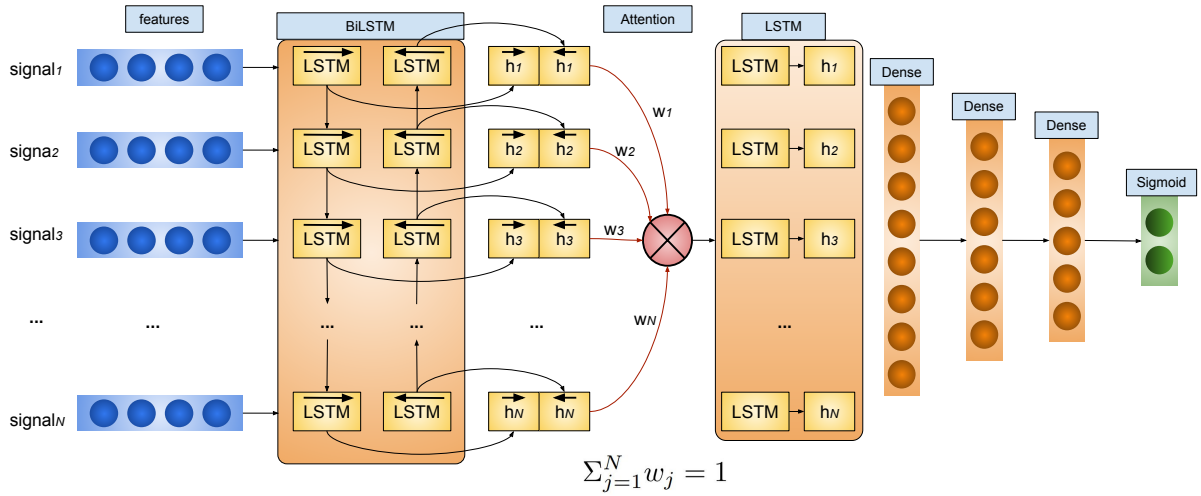


Figure 4: Architecture of the BiLSTM with Attention Mechanism.

- tp (true positive) is the number of males that are predicted as males.
- tn (true negative) is the number of females that are predicted as females.
- fp (false positive) is the number of females examples that are predicted as males.
- fn (false negative) is the number of males examples that are predicted as females.

4 Results and Discussions

This section discusses model performance results based on accuracy, F_1 score and binary cross entropy. The dataset is splitted into 90% for training, 10% for testing. The model is trained for 200 epochs and involved 3884 samples for training and 432 samples for testing.

4.1 Performance

Table 1 shows the testing results after evaluating the model. We observe BiLSTM obtaining high accuracy and F_1 score of 99% after 200 epochs. The BiLSTM outperformed the neural network models in (Harb and Chen, 2003; Azghadi et al., 2007; Meena et al., 2013; Qawaqneh et al., 2017a,b). Even though Qawaqneh et al. (2017a) used both images + audio files, their performance does not beat the BiLSTM. Figure 5a shows the accuracy curve of the BiLSTM model. The accuracy of model increased as the number of epochs increase.

Table 1: Comparison with other models

Model	Accuracy
MLP (Harb and Chen, 2003)	92
MLP (Azghadi et al., 2007)	96
ANN + Fuzy Logic (Meena et al., 2013)	65
DNN (Qawaqneh et al., 2017a)	64
DNN (Qawaqneh et al., 2017b)	59
BiLSTM-Attention	**99**

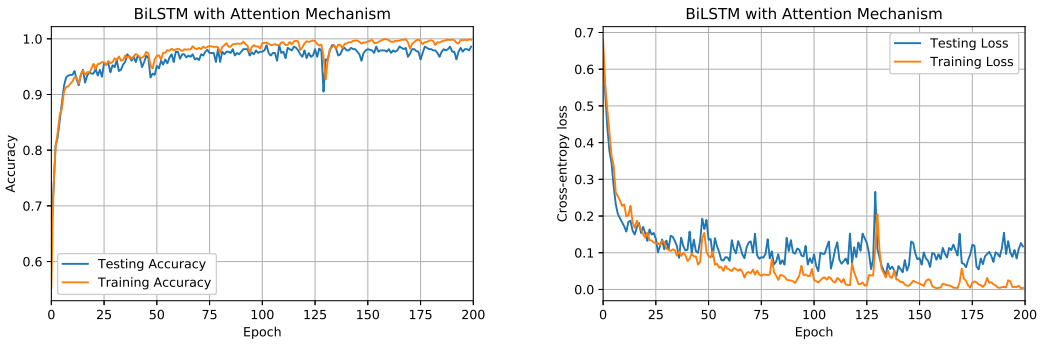
4.2 Overfitting

Overfitting happens when a model attempts to predict a trend in a noisy data. Overfitting is the consequence of a complicated model with excessive parameters. An overfitted model makes incorrect predictions as the trend does not represent the reality of the data. To show that overfitting is avoided, Figure 5b shows the binary cross entropy loss function curve. The loss function kept decreasing as number of training iterations increased. We observe BiLSTM reaching the lowest loss of 0.1 after 200 epochs. Hence, the model did not overfit.

5 Conclusion

This paper presented a Yorùbá gender recognition from speech using BiLSTM with attention mechanism. We discussed the literature on gender recognition. The acoustic features were explained together with normalization method. We explained the architecture of the proposed model. We observed BiLSTM achieving the state-of-the-art accuracy of 99% for a low-resourced language.

The future work will focus on using transformer models for gender recognition.



(a) Accuracy

(b) Binary cross entropy

Figure 5: Model prediction Accuracy and estimated binary cross entropy for BiLSTM.

References

Musaed Alhussein, Zulfiqar Ali, Muhammad Imran, and Wadood Abdul. 2016. Automatic gender detection based on characteristics of vocal folds for mobile healthcare system. *Mobile Information Systems*, 2016.

Abdul Wahab Funsho Atanda, Shahrul Azmi Mohd Yusof, and M Hariharan. 2013. Yorùbá automatic speech recognition: A review. In *Rural ICT Development (RICTD) International Conference*, pages 116–121.

S Mostafa Rahimi Azghadi, M Reza Bonyadi, and Hamed Shahhosseini. 2007. Gender classification based on feedforward backpropagation neural network. In *Artificial Intelligence and Innovations 2007: from Theory to Applications*, pages 299–304, Boston, MA. Springer US.

D. Bales, P. A. Tarazaga, M. Kasarda, D. Batra, A. G. Woolard, J. D. Poston, and V. V. N. S. Malladi. 2016. Gender classification of walkers via underfloor accelerometer measurements. *IEEE Internet of Things Journal*, 3(6):1259–1266.

Laurent Besacier, Etienne Barnard, Alexey Karpov, and Tanja Schultz. 2014. Automatic speech recognition for under-resourced languages: A survey. *Speech Communication*, 56:85–100.

S. Chaudhary and D. K. Sharma. 2018. Gender identification based on voice signal characteristics. In *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pages 869–874, Greater Noida (UP), India.

C. Chen, P. Lu, M. Hsia, J. Ke, and O. T. . Chen. 2011. Gender-to-age hierarchical recognition for speech. In *2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1–4.

J. Chen, S. Liu, and Z. Chen. 2017. Gender classification in live videos. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1602–1606.

Chris Ding and Xiaofeng He. 2004. K-means clustering via principal component analysis. In *Proceedings of the twenty-first international conference on Machine learning*, page 29. ACM.

Zhengming Ding and Yanjiao Ma. 2011. Manifold-based face gender recognition for video. In *Proceedings of 2011 International Conference on Computer Science and Network Technology*, volume 2, pages 1104–1107.

R. Djemili, H. Bourouba, and M. C. A. Korba. 2012. A speech signal based gender identification system using four classifiers. In *2012 International Conference on Multimedia Computing and Systems*, pages 184–187.

Theodoros Giannakopoulos. 2015. pyaudioanalysis: An open-source Python library for audio signal analysis. *PloS one*, 10(12):1–17.

Abdülkadir Gümüşçü, Kerim Karadağ, Mustafa Çalişkan, Mehmet Emin Tenekecı, and Dursun Akaslan. 2018. Gender classification via wearable gait analysis sensor. In *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4. IEEE.

H. Harb and Liming Chen. 2003. Gender identification using a general audio classifier. In *2003 International Conference on Multimedia and Expo. ICME '03. Proceedings (Cat. No.03TH8698)*, volume 2, pages II–733, Baltimore, MD, USA.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

W. Hwang, H. Ren, H. Kim, S. Kee, and J. Kim. 2009. Face recognition using gender information. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 4129–4132.

S. Jadav. 2018. Voice-based gender identification using machine learning. In *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, pages 1–4.

S. Kumar, S. Singh, and J. Kumar. 2019. Gender classification using machine learning with multi-feature method. In *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0648–0653.

P. J. Manamela, M. J. Manamela, T. I. Modipa, T. J. Sefara, and T. B. Mokgonyane. 2018. The automatic recognition of Sepedi speech emotions based on machine learning algorithms. In *2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD)*, pages 1–7.

Kunjithapatham Meena, Kulumani Subramaniam, and Muthusamy Gomathy. 2013. Gender classification in speech recognition using fuzzy logic and neural network. *International Arab Journal of Information Technology (IAJIT)*, 10(5).

T. I. Modipa and M. H. Davel. 2015. Predicting vowel substitution in code-switched speech. In *2015 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)*, pages 154–159.

B. Moghaddam and Ming-Hsuan Yang. 2000. Gender classification with support vector machines. In *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, pages 306–311.

T. B. Mokgonyane, T. J. Sefara, T. I. Modipa, M. M. Mogale, M. J. Manamela, and P. J. Manamela. 2019. Automatic speaker recognition system based on machine learning algorithms. In *2019 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA)*, pages 141–146.

B. Moore. 1981. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, 26(1):17–32.

Arjun Mukherjee and Bing Liu. 2010. Improving gender classification of blog authors. In *Proceedings of the 2010 conference on Empirical Methods in natural Language Processing*, pages 207–217. Association for Computational Linguistics.

Daniel van Niekerk, Etienne Barnard, Oluwapelumi Giwa, and Azeez Sosimi. 2015. Lagos-NWU Yoruba speech corpus.

Douglas Pulleyblank et al. 2017. Yoruba. In *The World's Major Languages*, pages 882–898. Routledge.

Zakariya Qawaqneh, Arafat Abu Mallouh, and Buket D. Barkana. 2017a. Age and gender classification from speech and face images by jointly fine-tuned deep neural networks. *Expert Systems with Applications*, 85:76–86.

Zakariya Qawaqneh, Arafat Abu Mallouh, and Buket D. Barkana. 2017b. Deep neural network framework and transformed MFCCs for speaker's age and gender classification. *Knowledge-Based Systems*, 115:5–14.

T. J. Sefara, M. J. Manamela, and P. T. Malatji. 2016. Text-based language identification for some of the under-resourced languages of South Africa. In *2016 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, pages 303–307.

T. J. Sefara, T. B. Mokgonyane, M. J. Manamela, and T. I. Modipa. 2019. HMM-based speech synthesis system incorporated with language identification for low-resourced languages. In *International Conference on Advances in Big Data, Computing and Data Communication Systems*.

Tshephisho Sefara, Promise Malatji, and Madimetja Manamela. 2016. Speech synthesis applied to basic mathematics as a language. In *South Africa International Conference on Educational Technologies*, pages 243–253.

Tshephisho Joseph Sefara and Madimetja Jonas Manamela. 2016. The development of local synthetic voices for an automatic pronunciation assistant. In *Southern Africa Telecommunication Networks and Applications Conference (SATNAC)*.

Tshephisho Joseph Sefara, Madimetja Jonas Manamela, and Thipe Isaiah Modipa. 2017. Web-based automatic pronunciation assistant. In *Southern Africa Telecommunication Networks and Applications Conference (SATNAC)*, pages 112–117.

Daniel Van Niekerk and Etienne Barnard. 2012. Tone realisation in a Yorùbá speech recognition corpus. In *Third Workshop on Spoken Language Technologies for Under-resourced Languages*, pages 54–59, Cape Town, South Africa.

MorphoBERT: a Persian NER System with BERT and Morphological Analysis

Mahdi Mohseni
Jena University Hospital
University of Jena
Jena, Germany
mahdi.mohseni@uni-jena.de

Amirhossein Tebbifakhr
Fondazione Bruno Kessler
University of Trento
Trento, Italy
atebbifakhr@fbk.eu

Abstract

Named Entity refers to person, organization and location names, and sometimes date, time, money and percent expressions as well. Named entity Recognition (NER) systems are developed to extract these essential information units from a text. Persian is a less-developed language in many natural language processing tasks such as NER. In this paper we present our system, MorphoBERT, submitted to the First Workshop on NLP Solutions for Under Resourced Languages (NSURL 2019)(Taghizadeh et al., 2019). We train the BERT model (Devlin et al., 2019a) on a large volume of Persian texts to get a highly accurate representation of tokens and then we apply a BiLSTM (bidirectional LSTM) on vector representations to label tokens. Persian is a rich language in terms of morphology and word parts may convey grammatical and semantic information. To inform the model of this information we analyze texts morphologically to split the lemma and affix(es) of each word and then we train the model on the analyzed texts. The test data, provided by the organizers, contains in-domain and out-of-domain texts. Our system achieves the first rank among all participated systems with a total high precision, recall and F1 of 87.0, 83.8, 85.4, respectively.

1 Introduction

Named Entity Recognition is a well-known classification topic in the research areas of language processing. NER systems aim to classify tokens of a text into classes such as person, organization and location, which are the most important named entity categories. Numerical expressions such as date, time, percent and monetary values are the other important classes, which are recognized in some systems. Named entities are the essential units of a text because either they convey most important information of the text or the text talks about them.

Various approaches have been used to recognize named entities in a text. Hidden Markov Model (Bikel et al., 1997), Maximum Entropy Model (Borthwick et al., 1998) and Conditional Random Filed (McCallum and Li, 2003) are statistical methods applied to the NER task. Neural network models have been also developed to categorize named entities. Collobert et al. (Collobert et al., 2011) propose a neural model based on a feed-forward architecture that takes into account a window of words around each target word. In (dos Santos and Guimarães, 2015) a convolutional neural network (CNN) is used to extract character-level and word-level embeddings representing contextual and structural word features. Ref. (Chiu and Nichols, 2016) combines a CNN model with a LSTM to utilize the strengths of both models. Lample et al. (Lample et al., 2016) approach the NER task using a hybrid statistical and neural model. In their model, LSTM-CRF, a bidirectional neural model extracts features from a text and CRF labels tokens.

Some research has focused on NER in Persian texts. PersoNER (Poostchi et al., 2016) is a Persian NER system in which a word embedding model and a sequential max-margin classifier are used. In (Poostchi et al., 2018) the LSTM-CRF model developed by (Lample et al., 2016) is applied to Persian texts. Shahshahani et al. (Shahshahani et al., 2019) have recently published a study in which a rule-based system, a CRF model and a LSTM-based system are compared on a newly well-designed Persian NER dataset.

After almost three decades of study, NER is still an open problem, especially for low-resource and under-developed languages such as Persian. The First Workshop on NLP Solutions for Under Resourced Languages (NSURL 2019) allocated a track to the Persian NER task (Taghizadeh et al., 2019). This paper presents our system called Mor-

phoBERT submitted to the workshop. In our system, we combine the BERT model (Devlin et al., 2019a) and a BiLSTM model and utilize a morphological analyzer developed for the Persian language. We train the BERT model on a large volume of Persian texts to get a highly accurate representation vector for each token in the input text. The Persian language is a morphologically rich language in which a single lemma may appear in various forms in a text. To allow the model to learn grammatical and semantic roles of lemmas and affixes, we first analyze words and split them into their constitutes. Then we feed them to the BERT model to generate a dense vector representation for them. Afterwards, a BiLSTM network gets the representations generated by BERT and tags tokens with the named entity labels.

Section 2 opens a discussion about the morphology of the Persian language and then explains our morphological analyzer. In Section 3, we describe our Persian NER Model. Section 4 covers resources that we use for training and evaluation of our system and presents the results under various experimental settings.

2 Morphological Analysis

Persian is an agglutinative language in which affixes and clitics attach to the base form of words. Not only verbs are inflected in the Persian language but also nouns and adjectives are highly affected by morphological rules of the language. Other part-of-speeches such as pronouns and adverbs may also get inflected especially in colloquial use. The main word order in Persian is subject-object-verb (SOV). The Persian script has an Arabic root and is written from right to left. In Persian texts short vowels are rarely written. It adds an ambiguity to processing a text as it produces many non-lexical homographs (Bijankhan et al., 2011), inflected words with the same spelling but different meanings and pronunciations.

There are several tenses in the Persian language and each verb is inflected in six different forms according to the person and number of the subject. Persian is a genderless language in which there is no discrimination between male and female, neither in its grammar nor in referring words. Nouns appears in a text as singular or plural. There are a few suffixes which create plural nouns from singulars. Few of these suffixes have been imported from Arabic. There is no definite article in the Persian language. However, indefinite articles have been defined in the language. There is no real possessive pronoun in the language and possession is expressed by adding clitics to a noun or sometimes to an adjective when it accompanies the noun. Fig. 1 shows some sample Persian word and their morphology.

Word	Analysis	Translation
کتابهایش (ketɒ:bhɒ:jaš)	کتاب + ها + ی + ش	his/her books
کتابهای زیبایش (ketɒ:bhɒ:je zibɒ:jaš)	کتاب + ها +ی؛ زیبا + ی + ش	his/her beautiful books
رفته‌ام (rafteam)	رفت + ه + ام	[I] have gone
می‌روم (miravam)	می + رو + م	[I] go
مسئولین (maso:lin)	مسئول + ین	authorities

Figure 1: Sample Persian words and their analyses.

Paykare (Bijankhan et al., 2011) is a Persian corpus designed and developed based on the EAGLES guidelines (Leech and Wilson, 1999) to capture the complexity of the Persian morphology. It contains almost 10M words, which have been manually tagged under a hierarchical structure. Although words are categorized into 14 major categories, the tagset consists of 109 distinct tags. A combination of these tags is used to label each word of the corpus. For example, ”ketɒ:bhɒ:yaš” (his/her books) has been tagged with ”N, COM, PL, 3” which stands for Noun, Common, Plural, 3rd possessive pronoun. The total number of hierarchical tags of words in the corpus rises up to 606 tags. We use this corpus to develop a Persian morphological analyzer.

Developing a morphological analyzer for Persian is very challenging. On the one hand, the Persian morphology is complex and ambiguous and requires an intensive contextual interpretation. On the other hand, some words have a beginning or an ending similar to affixes and clitics that makes the analysis error-prone. If a text could be tagged with the hierarchical tagging system of the Paykare corpus, one can analyze words precisely. But developing a fully automatic Part-of-Speech (POS) tagger with more 600 tags is demanding and a high accuracy is not achieved. We take another more practical approach. Texts in Paykare have been tagged manually, so, they are very accurate and reliable. As the hierarchical tag of a word reveals its structure and the way it has been created, one can develop a system to analyze words of the corpus. However, there are some exceptions which

need to be taken care of differently. For example, borrowed words from Arabic do not follow the Persian morphological rules and may be analyzed wrongly. For the exceptions, we make a list containing words and their correct analyses. The result shows that around 15.5% of the corpus consists of inflected words, which have different lemmas than their original surface forms in the texts.

Some words may have different analysis depending on their contexts. For each word and its major tag, we save the most frequent analysis in a map. For example, the word "ketɒ:bhɒ:yaš" (his/her book) with it major tag, "N", is anlyzed to "Ketɒ:b + hɒ: + y + aš". Tagging a text with only 14 major categories can be accomplished with a high accuracy. For a new text, we label the text with the major POS tags and then search the map to find the analysis of each word. To tag a new text with major POS tags, we use the Persianp toolbox (Mohseni et al., 2018). Since we take only the major tags into account we lose some information and cannot analyze all words correctly. However, the accuracy of the method remains very high. Using this method, only 3% of inflected words in the Paykare corpus are analyzed incorrectly. We use this method to analyze texts before training our NER neural model for the Persian language.

3 Persian NER Model

Our Persian NER system is depicted in Fig. 2. The lower layer is the morphological analyzer. Inflection changes the surface form of words and makes it difficult for a machine learning method to infer the role of words precisely and find out the grammatical and semantic role of lemmas and affixes. To help the model infer this information, words are split into their constitutes in the first layer. The neural part of the model is composed of the BERT model and a BiLSTM which are described below.

3.1 BERT

We use BERT (Devlin et al., 2019b) as a pre-training step. BERT is a language representation model in which bidirectional Transformer (Vaswani et al., 2017) is used in each layer of the model. It is trained by predicting masked words in an input sentence according to the preceding and proceeding words. This model can be trained on large-scale monolingual corpora. One of the advantages of using BERT compared to the word-level approaches such as word2vec

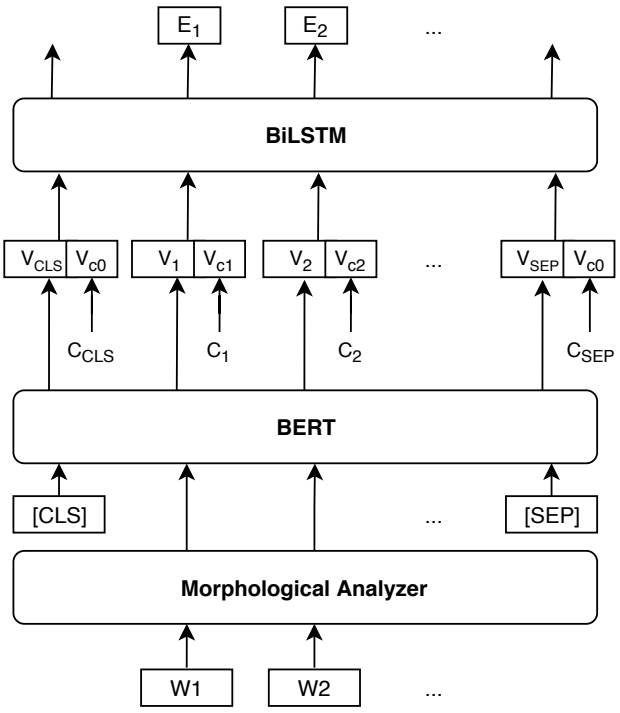


Figure 2: The architecture of the MorphoBERT NER system.

(Mikolov et al., 2013) and Glove (Pennington et al., 2014) is that the representation of each word is not fixed and is influenced by the other words in the sentence.

In our model, we use $\text{BERT}_{\text{BASE}}$, which uses 12 layers of bidirectional Transformer with 12 attention heads and 768 as the hidden size units. We pre-train the model from the scratch. This allows us to use our strategy in analyzing word morphologically and paying attention to the language-specific features. Also, we train the model on Persian monolingual data to have a more accurate model, while the available pre-trained models are multilingual and may have less performance on representing the Persian texts.

As depicted in Fig. 2, the output of the morphological analyzer is delivered to the BERT model. $[CLS]$ and $[SEP]$ are two tokens added by BERT to each input sentence indicating its boundary.

3.2 BiLSTM

We use a bidirectional LSTM to tag the named entities in sentences. As it is shown in Fig. 2, the input of this network is the representation of the sentence obtained from the BERT model. We use a bidirectional LSTM in order to leverage both left and right context to tag tokens. On the top of the BiLSTM, we use a linear model with the Softmax

activation function to get the probability distribution over all tags for each token.

3.3 Word Class Feature

Although neural models are very successful in extracting contextual information from a text, providing explicit features can still improve their performance. In Ref. (Shahshahani et al., 2019) that a LSTM-based model is proposed for Persian NER, feeding a feature representing word clusters enhances the result. This feature is the cluster number of the word, which is given to the model as another input. As Fig. 2 shows, we take the same approach and give the word cluster feature to the BiLSMT network. The representation of words generated by BERT is not fixed, so, we train a word2vec model to get a fix representation for each word. Then we apply a k-means clustering on word vectors. The number of clusters is set to 1500. The distance between instances is computed using cosine similarity. To create the word2vec model and cluster words we use the Gensim library [1]. The cluster number of each word is fed to the model and a cluster number is reserved for unknown words. The cluster numbers have their own embedding vectors, which are learned during training. The size of the vectors is set to 32. The cluster representation and the representation generated by BERT for each token are concatenated into a 800-dimensional vector and is given to the BiLSTM. Our experiments show that adding this feature improves the F1 measure of the system by 0.5%.

4 Experiments

4.1 Unlabeled Text Corpus

To train the BERT model for the Persian language, we collected a large volume of Persian texts consisting of news articles and Wikipedia documents. News articles crawled from 10 online news agencies contain 300M words and the dump of Persian Wikipedia[2] provides texts with more than 75M words. All texts are analyzed with our morphological analyzers and fed to the BERT model. We trained the model with more than 1M steps with the batch size equal to 16. The max sequence length of input sentences is set to 256 and the values of the parameters for masking words is set to the default values i.e. 15%. We use the Adam optimizer with initial learning rate equal to 5×10^{-5} and 10,000 warm-up steps. The vocabulary contains words with frequency more than 80 and its size reaches to 52K tokens.

[1] https://radimrehurek.com/gensim/index.html
[2] https://dumps.wikimedia.org/fawiki/latest/

Table 1: The statistics of the training dataset.

Named Entity	#Entities (phrases)	#Words
Person	12553	21121
Organization	14285	34774
Location	15412	21102
Date	4474	10413
Time	572	1786
Money	1295	4726
Percent	12557	2386
Total	49592	96308

Table 2: The statistics of the test dataset.

	No. Words
In-domain	68063
Out-of-domain	76463
Total	144526

4.2 NER Dataset

The organizers of the Persian NER task in NSURL 2019 have provided a training dataset and the final assessment on the test dataset is blind. The main features of the dataset have been described in (Shahshahani et al., 2019). The provided dataset has a similar structure to the CoNLL format in which each line contains one single word and its label separated by a $< TAB >$. The format of labels are IOB. The dataset contains almost 900K words from which about 50K are named entities. 7 types of entities tagged in the dataset are *person*, *organization*, *location*, *date*, *time*, *money* and *percent*. Table 1 presents the number of entities and the number of words in entity phrases.

The test dataset contains in-domain and out-of-domain texts. Table 2 show the size of the each part. Since the evaluation on the test dataset is blind we do not know the number of named entities in the dataset.

Table 3: The detailed results of MorphoBERT on provided dataset using 5-fold cross validation at the phrase-level for both subtasks.

Subtask	P	R	F1
3-class	87.2	89.2	88.2
7-class	86.2	88.5	87.4

4.3 Results

Once the BERT model is trained with the unlabeled text corpus, its output, the representation vectors of input tokens, is supplied to the BiLSTM network. As previously mentioned, word clusters are also given to the BiLSTM network as an extra features. We apply the same optimization method here as we did for training BERT. We don't fix the parameters of BERT allowing them to be fine-tuned. The number of epochs and the batch size are set to 10 and 32, respectively.

In the Persian NER task of NSURL 2019 (Taghizadeh et al., 2019), two subtasks have been defined. The first one is 3-class Persian NER in which 3 major named entities, *person*, *organization* and *location* are detected. The second subtask, called 7-class Persian NER, takes all types of entities in the dataset into account.

We first report the performance of our Persian NER system, MorphoBERT, on the provided dataset with 5-fold cross validation. Table 3 shows the results of system for both 3-class and 7-class subtasks at the phrase-level.

Table 4 presents the detailed results for all named entities. The evaluation at the word-level, which is obviously higher than the phrase-level, is presented in 5. In the table 'B-' and 'I-', corresponding to the IOB format, indicate respectively the beginning word and the inside word(s) of a named entity. The performance of the system on 3 main classes of *person*, *organization* and *location* is very high. *Percent* and *money* are phrased in a text in a relatively low number of predefined templates and they can be classified with a high precision and recall. In *date* and *time* the performance is lower. This is because of two reasons. First, in the dataset the number of instances for these two types of entities are low, so, the system cannot learn these classes very well. Second, according to the guideline of the dataset temporal phrases are labeled as entities when they are not generic and can be exactly specified knowing the the production time of the document. Therefore, it is very challenging for the system to discriminate between generic and specific temporal expressions. Comparing Table 4 and 5 shows that the results of the system at the word-level is higher as it is expected. It also states that it is more challenging to detect the correct boundary of some entities such as *location*. *I-Location* shows the inside word(s) of location entities. There is about 10% difference in performance between *B-Location* and *I-Location* tags. In Persian many location names are multiword and sometimes they cannot be inferred very well from pre-known instances. Using a rich gazetteer can alleviate this problem.

The organizers of the task evaluated the participated system in both subtasks on the test dataset.

Table 4: The detailed results of MorphoBERT on the provided dataset using 5-fold cross validation at the phrase-level.

Named Entity	P	R	F1
Person	91.5	91.4	91.5
Organization	94.2	88.0	90.9
Location	88.3	90.2	89.3
Date	77.1	82.0	79.5
Time	66.5	75.4	70.7
Money	89.9	93.1	91.5
Percent	94.2	88.0	90.9
Total	86.2	88.5	87.4

Table 5: The results of MorphoBERT on provided dataset using 5-fold cross validation at the word-level.

Named Entity	P	R	F1
B-Person	93.9	92.9	93.4
I-Person	94.1	94.2	94.1
B-Organization	87.3	89.6	88.4
I-Organization	91.8	89.2	90.5
B-Location	91.0	91.7	91.4
I-Location	84.5	77.0	80.6
B-Date	82.8	84.5	83.6
I-Date	87.5	86.9	87.2
B-Time	77.0	79.9	78.4
I-Time	80.8	85.2	82.7
B-Money	94.2	96.3	95.2
I-Money	96.8	97.5	97.2
B-Percent	95.9	89.0	92.2
I-Percent	97.7	95.9	96.8
Total	90.5	89.8	90.2

Table 6: The results of MorphoBERT on the test dataset at the phrase-level. (In: in-domain, Out: out-of-domain)

	3-class Subtask			7-class Subtask		
	P	**R**	**F1**	**P**	**R**	**F1**
In	88.7	85.5	87.1	88.4	84.8	86.6
Out	86.3	83.8	85.0	86.0	83.1	84.5
Total	87.3	84.5	85.9	87.0	83.8	85.4

Table 7: The results of MorphoBERT on the test dataset at the word-level. (In: in-domain, Out: out-of-domain)

	3-class Subtask			7-class Subtask		
	P	**R**	**F1**	**P**	**R**	**F1**
In	92.5	86.7	89.5	94.0	89.1	91.5
Out	91.5	84.0	87.6	91.8	85.7	88.6
Total	92.1	85.2	88.5	92.8	87.1	89.9

Our system, MorphoBERT, gained the first rank among the participated teams in all evaluation measures, in both tasks, and in in-domain and out-of-domain data.

Table 6 and 7 present the results of our system at the phrase-level and word-level, respectively. Comparing the results of the system on the in-domain test data with results of the system on the provided dataset (Table 3) shows that the precision remains high but the recall decreases. This reveals that the coverage of texts in the in-domain part of the test dataset is slightly different from the provided dataset, though the domain is the same. In the out-of-domain data, the decrease in the precision is negligible. However, the recall declines more seriously, evidently because of the difference of named entities covered in different domains.

We do not have access to the gold labels of the dataset. However, in order to have a more comprehensive analysis, we present the detailed results of MorphoBERT on the test dataset reported by the organizers. As Table 8 shows, the most decrease happens in *organization* and it is more than 10%. This shows that the test dataset contains organizations which are not observed in the training dataset. Regarding this fact that more than half of the test dataset consists of out-of-domain text, one can concludes that they come mostly from out-of-domain texts. It is not surprising that if the domain changes, the text refers to different organization names. Other named entities such as *person*, *date* and *time* however experience less changes.

Table 8: The detailed results of MorphoBERT on the test dataset at the phrase-level.

Named Entity	**F1**
Person	90.4
Organization	80.3
Location	87.1
Date	78.9
Time	71.0
Money	93.6
Percent	96.8
Total	85.4

5 Conclusions

We participated in the Persian NER task of NSURL 2019 with our system called MorphoBERT. Our system achieved the first rank in all settings among the participated teams. The system benefited from the BERT model and a Persian morphological analyzer. The assessment on the test dataset was blind. The task had two subtasks, 3-class and 7-class subtasks, and the system was evaluated on the in-domain as well as out-of-domain data. On the in-domain test data the total performance of the system is comparable with the system trained on the provided dataset and it changes slightly. Differentiating between generic temporal expressions with specific ones was a big challenge for the system and as a result the system gained the lowest results in the *time* and *data* classes. Another reason for getting a lower performance in these two classes was the low number of instances in the training dataset. Utilizing a statistical or even a rule-based system might be helpful here. Results showed that on out-of-domain texts the recall of the NER system decreases more, especially in detecting *organization*. This gives us a hint to focus on this challenge for future work. It is also worth focusing on the morphological analyzer. Our current morphological analyzer is not highly accurate in low-frequent and unknown words. Developing a high precise Persian morphological analyzer can be beneficial for many tasks, especially if there are no enough resources available to train data-voracious neural systems.

References

Mahmood Bijankhan, Javad Sheykhzadegan, Mohammad Bahrani, and Masood Ghayoomi. 2011. Lessons from building a persian written corpus: Peykare. *Language Resources and Evaluation*, 45(2):143–164.

Daniel M. Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. 1997. Nymble: a high-performance learning name-finder. In *Fifth Conference on Applied Natural Language Processing*, pages 194–201, Washington, DC, USA. Association for Computational Linguistics.

Andrew Borthwick, John Sterling, Eugene Agichtein, and Ralph Grishman. 1998. NYU: Description of the MENE named entity system as used in MUC-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*.

Jason P.C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Geoffrey Leech and Andrew Wilson. 1999. Standards for tagsets. In *Syntactic wordclass tagging*, pages 55–80. Springer.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 188–191.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Mahdi Mohseni, Javad Ghofrani, and Heshaam Faili. 2018. Persianp: A persian text processing toolbox. In *Computational Linguistics and Intelligent Text Processing*, pages 75–87, Cham. Springer International Publishing.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Hanieh Poostchi, Ehsan Zare Borzeshi, Mohammad Abdous, and Massimo Piccardi. 2016. PersoNER: Persian named-entity recognition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3381–3389, Osaka, Japan. The COLING 2016 Organizing Committee.

Hanieh Poostchi, Ehsan Zare Borzeshi, and Massimo Piccardi. 2018. BiLSTM-CRF for Persian named-entity recognition ArmanPersoNERCorpus: the first entity-annotated Persian dataset. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).

Cícero dos Santos and Victor Guimarães. 2015. Boosting named entity recognition with neural character embeddings. In *Proceedings of the Fifth Named Entity Workshop*, pages 25–33, Beijing, China. Association for Computational Linguistics.

Mahsa Sadat Shahshahani, Mahdi Mohseni, Azadeh Shakery, and Heshaam Faili. 2019. PEYMA: A tagged corpus for Persian named entities. *Journal of Signal and Data Processing, Vol. 16, Issue 1, 06-2019*.

Nasrin Taghizadeh, Zeinab Borhani-fard, Melika Golestani-Pour, Mojgan Farhoodi, Maryam Mahmoudi, Masoumeh Azimzadeh, and Heshaam Faili. 2019. NSURL-2019 task 7: Named entity recognition (ner) in farsi. In *Proceedings of the first International Workshop on NLP Solutions for Under Resourced Languages*, NSURL '19, Trento, Italy.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

AtyNegar at NSURL-2019 Task 8: Semantic Question Similarity in Arabic

Atieh Sharifi, Hossein Hassanpoor, Najmeh Zare Maduyieh
Department of NLP, Dade Pardazi Shenakht Mehvar Atynegar (DSA) Institute
Tehran, Iran
sharifiatieh@gmail.com
{hassanpoor, zare}@atynegar.ir

Abstract

Measurement of semantic similarity plays an important role in many areas of natural language processing. Several approaches have been proposed to determine the similarity of sentences in different languages but many of them are not extendable in all languages. According to the complicated Arabic language structure and lack of necessary resources and tools, the Arabic semantic similarity measurement is challenging.

In this paper, we proposed a supervised method for Arabic semantic question similarity measurement. Forty-one features (lexical, syntactic and semantic) are extracted from two question phrases, then the best distinctive features are selected by using SelectKBest algorithm. Finally, for sentences classification and determining the similarity score, SVM used.

The system participated in task8 of NSURL 2019 .The results of using this method on the data set of NSURL 2019 have a F-measure of 82.58 percent, which have improved the basic method.

1 Introduction

Nowadays we encounter a massive amount of text data. Due to the ease of changing a text, similar data are produced abundantly. Measuring text similarity is useful in many cases such as information retrieval, text classification, document clustering, topic detection, question answering, essay scoring, short answer scoring and machine translation. Because of the expansion of text resources and various applications of finding similar texts, the importance of similarity detection can be clearly understood (Gomaa and Fahmy, 2013). As a result, using appropriate methods that can easily recognize similar texts is of great importance.

The most fundamental part in sentences similarity measurement is determining words similarity. Words can be similar both lexically or semantically. Two words are lexically similar if they have a similar character sequence. However semantically similar words used in the same cases, same context or one is a type of another. In this paper several string-based algorithms proposed to determine lexical similarity. Also some corpus-based algorithms proposed to determine semantic similarity (Gomaa and Fahmy, 2013).

String-based algorithms operate on string sequences while corpus-based algorithms determine the similarity between words according to information gained from a large corpus. One approach to measure similarity is using deep learning to represent words and texts as vectors. Similar words have closer vectors and dissimilar words have distant vectors. Therefore, words similarities can be determined by measuring words vector distances. In this paper in addition to words vector representation, we also use sentences vector representation.

In order to increase accuracy, word alignment and syntactic overlapping used to determine similarity. 41 features obtained for two sentences which 38 of them chosen as effective features and used to train the model. The system participated in task8 of NSURL 2019 (Seelawi et al., 2019).

This paper is organized as follows: Section two presents related works in this field, section three introduces the proposed approach and section four representing the results. Finally, section five contains conclusion and suggestions.

2 Related Works

During the last decade, several methods were established to measure sentence similarity based

on semantic, syntactic and statistic knowledge. In this section we introduce some related works in determining Arabic sentences and texts similarity.

Wali et al. (Wali et al., 2017) proposed a supervised approach in which three types of features, lexical, semantic and syntactico-semantic, are used to determine sentences similarity. Lexical feature computed based on common terms between the sentences and Jaccard coefficient. In computing semantic features, each sentence represented with a vector and then the cosine similarity of these two vectors are computed. The vectors created by forming a word set using only the distinct terms of the pair of sentences. If the term is in the sentence, the corresponding element in the vector, set to 1 and if the term isn't in the sentence, the corresponding vector element is equal to the highest similarity between the term and the words of the sentence. The similarity of two words calculated using the number of common synonyms of the two words based on LMF standardized dictionaries. The syntactico-semantic features also computed using these dictionaries and common semantic arguments between the pair of sentences. Finally Support Vector Machine used for regression. The F-measure of using this approach on gathered data is 85.6%.

Elghannam (Elghannam, 2016) computing Arabic texts similarity by their words similarity. Each word represented as a vector. This vector is a set of co-occurrence words extracted from a corpus. DISCO tool is used for this purpose. DISCO builds the second order word vectors by first counting words co-occurrences to build the co-occurrence matrix. Cosine similarity of two vectors shows the similarity between two words. The highest accuracy of this method on news data is 97%.

Nagoudi et al. (Nagoudi and Schwab, 2017) represents each word with a vector using word embedding. The vector of each text is the sum of its words vectors. The similarity of two texts computed using cosine similarity between texts vectors. To determine the importance of each word, the word IDF and the part of speech (each part of speech has a score) multiplies the word vector. Best result obtained by using syntactic template and the Pearson correlation is 79.69%.

Al-Smadi et al. (AL-Smadi et al., 2017) proposed a supervised approach to compute text similarity with lexical, semantic and alignment features. These include word overlap, POS tag n-grams overlap, NER overlap, Levenshtein similarity, words alignment and topic modeling (to recognize two texts with a same topic). Finally, a support vector machine used for regression. The results of this approach on news tweets has F-measure of 87.2%.

3 Proposed Method

As mentioned before, the proposed method is supervised approached including preprocessing, feature extraction and classification phases. The preprocessing phase includes removing diacritics, excess spaces, tatweel character and correcting punctuation spacing. In the feature extraction phase, 41 features (lexical, syntactic and semantic) are extracted from two question phrases, then the best distinctive features are selected by using SelectKBest algorithm. Then by classifying the sentences according to the best distinctive features, similarity of the questions is determined.

3.1 Feature Extraction

We consider a total of 41 features. These features explained below.

Words overlap: this type of features computed based on the number of common words in two sentences. These features obtained through the stems vectors of a sentence. In this step we perform tokenizing, then we remove stop words and punctuations. Finally, word stems compared with each other. These features computed for n-grams (n=1,2,3) and precision, recall and F-measure calculated for each of them (AL-Smadi et al., 2017; Karampatsis, 2015).

For n-grams (n=1, 2, 3), precision, recall and F-measure computed as below. If the denominators of the first and second relations are both zero, 1 is considered as the value of all three features. If one of the denominators or the numerator is zero, 0 considered as the value of all three features. These are true for POS tag overlap and NER overlap features too as is stated in the following sections.

$$lexicalP_n = \frac{num\ of\ common\ n-grams}{num\ of\ first\ sentence\ n-grams} \quad (1)$$

$$lexicalR_n = \frac{num\ of\ common\ n-grams}{num\ of\ second\ sentence\ n-grams} \quad (2)$$

$$lexicalF_n = \frac{lexicalP_n \times lexicalR_n}{lexicalP_n + lexicalR_n} \quad (3)$$

POS tag overlap: The syntactic similarity of the two sentences is obtained using the number of

common syntactic patterns. POS tag vectors of sentences is used for these features. In this step, we removed punctuation marks. After word-tokenizing, their POS tags is specified. Similar to the words overlap feature, the syntactic pattern overlap is computed for 1,2,3-grams, and for each, the accuracy, recall and the F-measure are calculated as follows (AL-Smadi et al., 2017).

$$posP_n = \frac{num\ of\ common\ POS\ tags}{num\ of\ first\ sentence\ POS\ tags} \quad (4)$$

$$posR_n = \frac{num\ of\ common\ POS\ tags}{num\ of\ second\ sentence\ POS\ tags} \quad (5)$$

$$posF_n = \frac{posP_n \times posR_n}{posP_n + posR_n} \quad (6)$$

Named entity overlap: 9 features obtained in this step. These features gathered using named entities vectors. At first we tokenize the sentence. Then the entities are specified. Similarity is also based on the type of named entity (place, person, organization) and the word itself. The number of common named entities is calculated for 1,2,3-grams, for which the precision, recall and F-measure are calculated as follows (AL-Smadi et al., 2017).

$$nerP_n = \frac{num\ of\ common\ NEs}{num\ of\ first\ sentence\ NEs} \quad (7)$$

$$nerR_n = \frac{num\ of\ common\ NEs}{num\ of\ second\ sentence\ NEs} \quad (8)$$

$$nerF_n = \frac{nerP_n \times nerR_n}{nerP_n + nerR_n} \quad (9)$$

Levenshtein maximum similarity: In this step, we calculate the similarity between two sentences based on their words similarity. These features are obtained using stems vectors of the sentences. First, we tokenize and remove punctuation marks. Then the words stems are compared. The Levenshtein method is used to determine the words similarity. A matrix of Levenshtein values is created for two sentences in which rows represent the stems of the first sentence, and the columns represent the stems of the second sentence. Then, by using this matrix, a vector created which contains the lowest values of each matrix row. In fact, we consider the words in the second sentence that are the most similar to the words in the first sentence, and store their Levenshtein values in the vector V. Then we sort this vector and keep only five minimum values. Precision, recall and F-measure are obtained in this step (AL-Smadi et al., 2017). These features are calculated using the sum of vector V values as follows:

$$levP_n = \frac{sum\ of\ V\ values}{num\ of\ first\ sentence\ words} \quad (10)$$

$$levR_n = \frac{sum\ of\ V\ values}{num\ of\ second\ sentence\ words} \quad (11)$$

$$levF_n = \frac{levenshteinP_n \times levenshteinR_n}{levenshteinP_n + levenshteinR_n} \quad (12)$$

Alignment: This group of features computed by the assumption that the two semantically similar sentences can be aligned. To compute these features, we use the V vectors which we obtained in the previous section. If W_i in the first sentence is the most similar word to W_j in the second sentence, |i-j| shows the value of W_i and W_j alignment. For each word in V, the alignment value computed and stored in vector Y (AL-Smadi et al., 2017). Then Precision, recall and F-measure calculated as follows:

$$alignmentP = \frac{sum\ of\ Y\ values}{num\ of\ first\ sentence\ words} \quad (13)$$

$$alignmentR = \frac{sum\ of\ Y\ values}{num\ of\ second\ sentence\ words} \quad (14)$$

$$alignmentF = \frac{alignmentP \times alignmentR}{alignmentP + alignmentR} \quad (15)$$

Character sequence: There are 4 features in this step, each of them divided to the minimum length of the two sentences. First we tokenize the sentences, then remove stop words and punctuation marks, after that we extract words stems and rebuild the sentences (Tian et al., 2017).

- LCPrefix: The largest common prefix of two sentences
- LCSuffix: The largest common suffix of two sentences
- LCSubString: The largest common substring of the two sentences
- LCSequence: The largest common sequence of two sentences. Here we consider the common characters in the two sentences.

BOW similarity: In this step, a vector is considered for each sentence. The cosine similarity of these two vectors is considered as a feature for the two sentences. Initially, we tokenize the sentences. Then the punctuation marks and stop words are removed from the tokens. The vectors of the two sentences have a same size which is equal to the size of the common unique words in these sentences. If the word in the vector exists in the sentence, the IDF of that word will replace it,

otherwise its value will be zero (Tian et al., 2017). The IDF values are created using the Arabic Wikipedia corpus.

Word embedding similarity: A simple definition for word embedding is to consider a vector of numbers for each word. The words that are more similar to each other, have closer vector space. This vector specifies the syntax, semantic and other features of the word. This way, it is possible to display each word in tens or hundreds of dimensions. There are several algorithms for this purpose, while FastText is used here (Grave et al., 2018). Arabic Wikipedia used to build this model. Using the FastText pertained model, you can get a 300-dimensional vector for each word. First, tokenization performed, then the punctuation marks removed. Each sentence is represented as a vector. This vector is obtained from the sum of the vectors of the sentence words and finally their average. Then a feature is obtained using the cosine similarity of these two vectors (Eyecioglu and Keller, 2016).

Word mover's distance (WMD) score: In this feature, the distance between two sentences is obtained based on the words vectors distance using FastText model. The more similar the words of two sentences, the less distance between sentences vectors. Thus for same sentences this value tends to zero. First, the sentences are tokenized, then the punctuation marks and stop words removed. Finally, the distance between two sentences is calculated.

Doc2vec similarity: in the FastText model, each word represented by a vector. Unlike FastText, the doc2vec model gives us a numerical representation of a document, and we use it here to construct a vector for each sentence. Arabic Wikipedia has been used to construct this model. First, we tokenize the sentence and remove the punctuation marks. By using the pre-trained model, a doc2vec vector created for each sentence. Then a feature is obtained using the cosine similarity of these vectors.

3.2 Sentence Classification

We use support vector machine to classify the sentences. After feature extraction and creating the training data, preprocessing these data should be done. As the first step we normalize the data. In the process of normalization, the values of each feature (each column) is mapped to zero-mean and unit variance values. Finally, the best features are selected. For this purpose, SelectKBest algorithm is used. This algorithm gets the number of selected features n as the input. Then the model is built using the selected features.

4 Evaluation

In order to select appropriate features, we examined different inputs for SelectKBest algorithm. All the training data of NSURL task8, was used for test and the training model is built using SVC. Due to Table 1, SelestKBest(38) algorithm has the best results for detecting similar sentences. Table 2 shows the score and the effect of each feature using this algorithm. These features are sorted by their score. According to this table, precision, recall and F-measure of NER 3-grams overlap are the three deleted features that have the least score. Because of the short sentences, there is no NER overlap at the 3-gram level in the sentences, so this feature is not effective.

The SVC model is built, using the 38 features mentioned before. The model is evaluated using the NSURL 2019 task8 test data. The F-measure of the proposed method is 82.58%.

Algorithm	F-score	Recall	precision
SelectKBest(37)	0.798	0.767	0.832
SelectKBest(38)	0.808	0.783	0.834
SelectKBest(39)	0.807	0.782	0.834
SelectKBest(41)	0.806	0.782	0.831

Table 1: Results of testing SelectKBest algorithms on svc model.

5 Conclusion and Suggestions

In this paper, in order to detect similar Arabic questions, a supervised approach proposed. 38 effective feature s are extracted for each pair of questions which contain syntactic, semantic and lexical features. Semantic features are obtained using word embedding and doc2vec. Lexical features are obtained by words overlap feature. String based algorithms and syntactic features are also obtained using the syntactic structure of the sentence.

Due to Table 2, words overlap is one of the most effective features. After that, the largest common suffix, Word mover's distance, Levenshtein similarity, doc2vec similarity and bag of words similarity have the highest priority respectively.

Feature	Score	Feature	Score	Feature	Score	Feature	Score	Feature	Score
lexicalP1	4095	levF	1723	alignmentR	879	nerR1	208	nerP2	19.7
lexicalF1	3055	Doc2vec	1564	alignmentF	646	nerP1	206	LCPrefix	6
LCSuffix	2871	levP	1488	posP1	633	Word_Embedding	204	nerP3	2
lexicalP2	2624	BOW	1276	posP2	380	posF3	146	nerR3	2
lexicalP3	2624	LCSequence	1239	posF1	326	posR2	60	nerF3	2
lexicalF2	2179	lexicalR2	1231	alignmentP	314	posR3	50		
lexicalF3	2179	lexicalR3	1231	posP3	229	posR1	33		
WMD	1885	LCSubString	998	posF2	224	nerR2	20		
levR	1822	lexicalR1	910	nerF1	210	nerF2	19.9		

Table 2: Features ranking using selectkbest algorithm.

Lexical features are effective because there is high word overlap between similar sentences. Also there are some synonyms in some of similar sentences, so Word mover's distance feature can be helpful in detecting such sentences. But sometimes instead of two words, two phrases can be equivalent in meaning. Such cases are harder to detect. In Doc2Vec feature the whole sentence represents as a vector so it can covers some of the flaws. In some pairs of questions, the meaning has changed with displacement of the words, although in many cases this does not change the meaning, so the alignment feature can be somewhat effective in identifying similar sentences.

After examining the sentences which were incorrectly identified as similar, we found that removing stop words improved the accuracy of the system, although in some cases deleting these words has led to a mistaken identification. For example, some question words like who or when are effective in similarity detection but some of these words are ignored by removing stop words. Also in some cases there is excess information in one of the sentences which doesn't change the meaning but leads to incorrect similarity detection.

In order to improve the results, we can also consider the similarity of the question words in the two sentences. For example, the question word "which year" is equivalent to "when". Also the synonym words in two sentences can be determined using the semantic networks. Then in computing words overlap we can assume that these words are equal. To identify the similarity between words, instead of Levenshtein similarity, semantic networks can be used (Pawar and Mago, 2018). Syntactic n-grams overlap using the sentence dependency tree is another feature that can be effective in determining the similarity of two sentences (Segura-Olivares et al., 2013; Kohail et al., 2017)

References

Mohammad AL-Smadi, Zain Jaradat, Mahmoud AL-Ayy, and Yaser Jararweh. 2017. Paraphrase identification and semantic text similarity analysis in Arabic news tweets using lexical, syntactic, and semantic features. *Information Processing and Management,* 53(3):640-652. https://doi.org/10.1016/j.ipm.2017.01.002.

Fatma Elghannam. 2016. Automatic Measurement of Semantic Similarity among Arabic Short Texts. *Communications on Applied Electronics (CAE), 6*(2):16-21. https://doi.org/10.5120/cae2016652430.

Asli Eyecioglu and Bill Keller. 2016. ASOBEK at SemEval-2016 Task 1: Sentence Representation with Character N-gram Embeddings for Semantic Textual Similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 736-740.

Wael Gomaa and Aly Fahmy. 2013. A survey of text similarity approaches. *International Journal of Computer Applications, 68*(13):13-18. https://doi.org/10.5120/11638-7118.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning Word Vectors for 157 Languages. In Proceedings of the International Conference on Language Resources and Evaluation. Pages 3483-3487.

Rafael Michael Karampatsis. 2015. CDTDS: Predicting Paraphrases in Twitter via Support Vector Regression. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 75–79.

Sarah Kohail, Amr Rekaby Salama, and Chris Biemann. 2017. STS-UHH at SemEval-2017 Task 1: Scoring Semantic Textual Similarity Using Supervised and Unsupervised Ensemble. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017).* Pages 175-179.

El Moatez Billah Nagoudi and Didier Schwab. 2017. Semantic Similarity of Arabic Sentences withWord

Embeddings. In *proceedings of the Third Arabic Natural Language Processing Workshop (WANLP)*, pages 18-24.

Atish Pawar and Vijay Mago. 2018. Calculating the similarity between words and sentences using a lexical database and corpus statistics. *arXiv:1802.05667*.

Haitham Seelawi, Ahmad Mustafa, Hesham Al-Bataineh, Wael Farhan, and Hussein T.Al-Natsheh. 2019. {NSURL}-2019 Task 8: Semantic Question Similarity in Arabic. In *Proceedings of the first International Workshop on NLP Solutions for Under Resourced Languages.* Trento, Italy.

Andrea Segura-Olivares, Alejandro Garc´ıa, and Hiram Calvo. 2013. Feature Analysis for Paraphrase Recognition and Textual Entailment. *Research in Computing Science, 70*:144-119.

Junfeng Tian, Zhiheng Zhou, Man Lan, and Yuanbin Wu. 2017. ECNU at SemEval-2017 Task 1: Leverage Kernel-based Traditional NLP features and Neural Networks to Build a Universal Model for Multilingual and Cross-lingual Semantic Textual Similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017)*, pages 191–197.

Wafa Wali, Bilel Gargouri, and Abdelmajid Ben Hamadou. 2017. Enhancing the sentence similarity measure by semantic and syntactico-semantic knowledge. *Vietnam Journal of Computer Science, 4*(1):51-60. https://doi.org/10.1007/s40595-016-0080-2.

Beheshti-NER: Persian named entity recognition Using BERT

Ehsan Taher*
NLP Research Laboratory
Shahid Beheshti University
Tehran, Iran
e.taher@mail.sbu.ac.ir

Seyed Abbas Hoseini*
NLP Research Laboratory
Shahid Beheshti University
Tehran, Iran
seyeda.hoseini@mail.sbu.ac.ir

Mehrnoush Shamsfard
NLP Research Laboratory
Shahid Beheshti University
Tehran, Iran
m-shams@sbu.ac.ir

Abstract

Named entity recognition is a natural language processing task to recognize and extract spans of text associated with named entities and classify them in semantic Categories.

Google BERT is a deep bidirectional language model, pre-trained on large corpora that can be fine-tuned to solve many NLP tasks such as question answering, named entity recognition, part of speech tagging and etc. In this paper, we use the pre-trained deep bidirectional network, BERT, to make a model for named entity recognition in Persian.

We also compare the results of our model with the previous state of the art results achieved on Persian NER. Our evaluation metric is CONLL 2003 score in two levels of word and phrase. This model achieved second place in NSURL-2019 task 7 competition which associated with NER for the Persian language. our results in this competition are 83.5 and 88.4 f1 CONLL score respectively in phrase and word level evaluation.

1 Introduction

in this paper we trained our model which is participated in NSURL-2019 task 7 competition (Taghizadeh et al., 2019) which associated with NER for the Persian language.

Named Entity Recognition (NER) is one of the important and basic tasks in natural language processing, assigning different parts of a text to suitable named entity categories.

There are several sets of named entity (NE) categories introduced and used in different NE tagged corpora as their tagsets. For example, Peyma's (Shahshahani et al., 2018) tagset consists of person, organization, location, date, money, percent, and time, while the Arman tagset (Poostchi et al.) contains person, organization, location, facility, product, and event.

NER is one of the key parts of many downstream applications in NLP, such as question answering (Aliod et al., 2006), information retrieval (Guo et al., 2009), and machine translation (Babych and Hartley, 2003). As a result, the performance of NER can affect the quality of a variety of downstream applications. Furthermore, this effect is more obvious in low-resource languages because in these languages due to lack of data and tagged corpora, usually applications are implemented in pipe-line architecture unlike other languages like English which prefer to use End-to-End solutions.

Preparing basic tools in under-resourced languages by high performance can be a good solution to such languages while we counter with lack of data issue for training such tools.

We have trained conditional random field on the top of pre-trained bidirectional transformer BERT. Delvin et al. (Devlin et al., 2019) introduced BERT as a pre-trained Bidirectional Transformer model for language understanding tasks. BERT achieved state of the art results in many tasks like question answering, language inference, and Named entity recognition.(Devlin et al., 2019)

The need for large tagged data is the main problem with the recent supervised methods such as deep learning. Transfer learning can help this problem for under-resourced languages. Word embeddings approach (Mikolov et al., 2013),(Bojanowski et al., 2016), (Joulin et al., 2017) and (Peters et al., 2018) are the first kind of trans-

*Equal contribution.

fer learning solutions. We use word embeddings for supervised tasks after we trained them unsupervised on large raw corpora of texts. By this, they can reduce the need for huge labeled data. They defer by BERT usually in many aspects like the fine-tuning step. After pretraining BERT on large row corpora, we fine-tune it for our specific supervised problem. While BERT tokenizes text by itself, it extracts contextualized embeddings for each token. BERT is pre-trained on 104 languages like Persian, and this is one of the big advantages of this model. Vaswani et al. (Vaswani et al., 2017) introduced transformer architecture and self-attention as an alternative for encoder-decoder recurrent neural networks architecture which could achieve state of the art results in English to German and English to France machine translation problem. Furthermore, the speed for training transformers is much less than recurrent neural networks in encoder-decoder architecture. CRF as a probabilistic model like hidden Markov model makes it possible to extract and consider structural dependencies among tags in data. While Encoders like BERT try to maximize likelihood by selecting best hidden representation while CRF maximizes likelihood by selecting best output tags. We achieved 88.4% CONLL F1 score in word-level and 83.5% CONLL F1 score in phrase-level evaluation on Peyma dataset. We won second place in NSURL-2019 task 7 (Taghizadeh et al., 2019) competition for NER task.

In section 2, we talk about previous methods for NER and solutions like transfer learning to deal with under-resourced languages. Section 3 describes BERT. Section 4 explains our model in more details, discussing the training and test phases. In section 5, we show the achieved results on experiments like evaluating our model on different datasets. Section 6 concludes the paper.

2 Related Work

This paper describes a deep learning method based on word embedding and transfer learning, for named entity recognition in Persian language. Thus, in this section we first discuss some related work on Persian NER, then some recent work on English NER, and then talk about some word embedding models which can be used in NER tasks via transfer learning.

Mortazavi and Shamsfard (Mortazavi and Shamsfard, 2009) used a rule-based system to extract named entities for Persian languages. It was one of the first implementations for NER in Persian while no datasets existed in that time for evaluation. Poostchi et al. (Poostchi et al.) introduced new annotated Persian named-entity recognition dataset named Arman. Arman contains 250,015 tokens and 7,682 sentences. Set of entity categories consists of person, organization (like banks and ministries), location, facility, product, and event. They also trained conditional random field with bidirectional LSTM on this dataset as a baseline. Shahshahani et al. (Shahshahani et al., 2018) introduced new annotated Persian named-entity recognition dataset called Peyma. Peyma contains 7,145 sentences, 302,530 tokens and 41,148 tokens with entity tags collected from 709 documents. Class distribution for both Peyma and Arman datasets are presented respectively in Fig.1 and Fig.2.

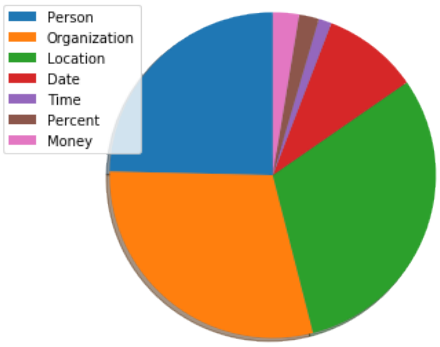


Figure 1: distribution of different classes in Peyma dataset

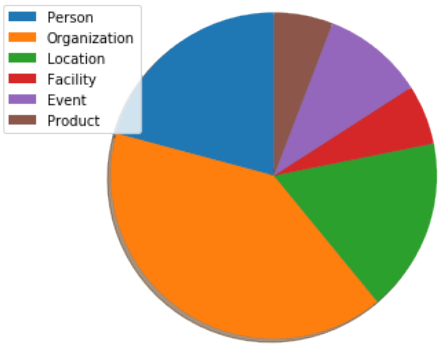


Figure 2: distribution of different classes in Arman dataset

Bokaei and Mahmoudi (Bokaei and Mahmoudi, 2018) trained recurrent and convolutional neural networks with CRF on Arman dataset.

Baevski et al. (Baevski et al., 2019) used a novel method for training bidirectional transformer which could over perform previous work and achieved state of the art result in English NER.

Akbik et al.(Akbik et al., 2018) used contextualized word embeddings extracted from character-level language model to solve the NER problem.

Delvin et al. (Devlin et al., 2019) introduced BERT as a pre-trained bidirectional transformer. They used and evaluated BERT on many tasks, including NER.

Using unsupervised methods can be a promising way because the most important issue for low resource languages is the lack of labeled data while but the access to a large amount of raw texts is more probable and feasible. Today, word embeddings such as Glove (Pennington et al., 2014), word2vec (Mikolov et al., 2013) , and fastText (Joulin et al., 2017) are essential parts of many methods in NLP. These models give continuous representations in n-dimensional space for each word which contain semantic information and features about that word.

Elmo (Peters et al., 2018) introduced deep contextualized word embedding by considering the context of words. Which means words have different embedding in different contexts. Delvin et al. (Devlin et al., 2019) and Radford et al.(Radford et al.) proposed a new method with transfer self-attention blocks without the need to change in architecture for a specific problem. They suggest fine-tuning pre-trained bidirectional transformers for specific problems.

Radford et al. (Radford et al.) introduced a new language model called GPT.2, which could reach 55% F1 score on the CoQA dataset without any labeled data. This approach tries to remove the need for labeled data and gives a general model to solve problems against BERT, which tries just to give a general model.

best performing models before us for NER in Persian are LSTM based models which usually come with CRF and pre-trained non-contextualized embedding layers. these models are evaluated on two common datasets for NER: PEYMA and ARMAN. Bokaei and Mahmoudi (Bokaei and Mahmoudi, 2018) and Shahshahani et al. (Shahshahani et al., 2018) had reported the best results which you can see in Table 3

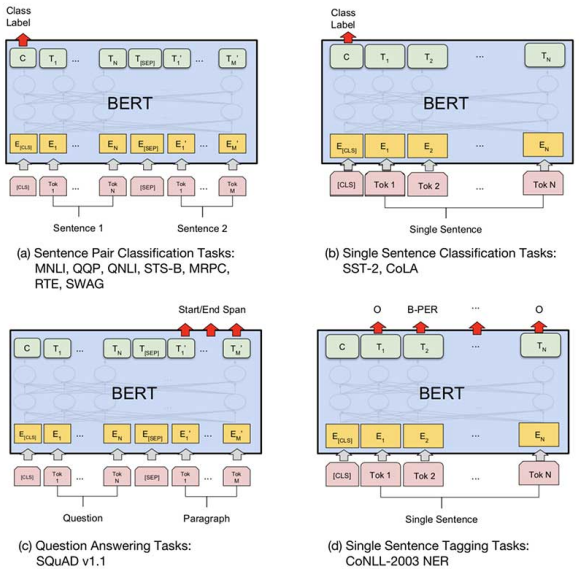


Figure 3: fine-tuning BERT in different tasks (Devlin et al., 2019)

3 BERT

BERT (Bidirectional Encoder Representations from Transformer) is a language model representation based on self-attention blocks. BERT is pre-trained in different language model tasks on raw unlabeled texts. The pre-trained deep bidirectional model with one output layer can reach state-of-the-art results in many tasks such as question answering and Multi-Genre Natural Language Inference. The idea is to have a general architecture which fits many problems and a pre-trained model which minimize the need for labeled data. For example, in Fig. 3 You see how BERT can be used in different tasks like question answering, sentences pair classification, single sentence classification, and single sentence tagging task. While each task has a different format of inputs and outputs. As mentioned before, one of the big advantages of BERT is that it was trained in 104 languages and Persian is one of those. Which motivate us to use it for NER in Persian.

4 Our Proposed Model

In this paper, we propose a method for Persian NER. In this method, we use BERT pre-trained model. As in NER task, we need to assign the most suitable tag to each token, and suitable tokenization is an important step.

While BERT has its tokenization with Byte-Pair encoding and it will assign tags to its extracted tokens, we should take care of this issue. BERT

extracted tokens are always equal to or smaller than our main tokens (that taken from the Step-1 Shamsfard et al., 2010) because BERT takes tokens of our dataset one by one. As a result, we will have intra-tokens which take X tag (meaning don't mention). We trained a conditional random field and fully connected layer after output representation of tokens extracted by BERT. It Is a fine-tuning step to make the entire model ready for NER task. You can see a schema of the model in Fig.4 .

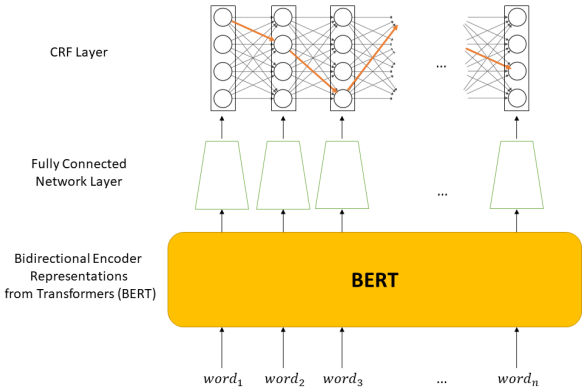


Figure 4: architecture of our trained model

5 Experiments

We have trained and tested our model on two different datasets: Peyma (Shahshahani et al., 2018) and Arman (Poostchi et al.).We split PEYMA dataset into 5 equal subsets (Peyma contains 7145 sentences thus each subset contains 1429 sentences) and use 5-fold cross-validation. we repeated training phase 5 times separately, Each time, one of the 5 subsets is used as the test set and the remaining 4 subsets are put together to form a training set. In all experiments, CONLL F1 score is calculated in two levels: word and phrase as a metric for evaluating the performance of model. Results of our model on Peyma and Arman datasets are given respectively in Table 1 And Table 2.

On Peyma dataset We can reach 90.59% CONLL F1 score in phrase-level and 87.62% F1 score in word level. Best results are seen for Percent class and worst for Time.

On Arman dataset, We reached 79.93% CONLL F1 score in phrase-level and 84.03% F1 score on word-level. Best results are seen for Person class and worst for Event

One of the causes for achieving different results in each class is the amount of named entities in the datasets. As can be seen in Fig.1 and Fig.2, the number of phrases for Time and Event classes are much lower than others.

As you see in Table 3 in both word and phrase levels, our model outperform other NER approaches for the Persian language. Unfortunately previous works reported their results just on one of the datasets. Shahshahani et al.(Shahshahani et al., 2018) reported their results just in word level evaluation on Peyma dataset.Table 3 shows that our results are 10 percent better than Shahshahani and colleagues on the same platform. On the other hand Bokaei and Mahmoudi (Bokaei and Mahmoudi, 2018) reported their results on Arman dataset Which is lower than ours in both word and phrase levels according to Table 3.

	Arman		Peyma	
	word	phrase	word	phrase
Bokaei and Mahmoudi (Bokaei and Mahmoudi, 2018)	81.50	76.79	-	-
Shahshahani et al.(Shahshahani et al., 2018)	-	-	80.0	-
Beheshti-NER (Our Model)	**84.03**	**79.93**	**90.59**	**87.62**

Table 3: comparing results of our trained model with others

The results of NSURL task-7 competition is reported in two levels of evaluation, namely word and phrase levels for two subtasks: A) NER for 3- classes (Person, Location, Organization) and B) NER for all classes. for the competition, we have trained our model on PEYMA corpus in addition to another corpus which was prepared by Iran Telecommunication Research Center (ITRC). The organizers also used two kinds of in-domain and out-domain test data. Our model won second place in all of these evaluation types.

Tables 4, 5, 6, 7 and 8 show the results of evaluation reported by competition for all teams which participated in the challenge. Our method is mentioned as Beheshti-NER-1[1]. Table 4 and 5 show the results for subtask A. according to the tables, we reached to 84.0% and 87.9% F1 score respectively for phrase and word level evaluations.

[1]Code is available at https://github.com/sEhsanTaher/Beheshti-NER

	Date		Location		Money		Organization		Percent		Person		Time		
	B-	I-	B-	I-	B-	I-	B-	I-	B-	I-	B-	I-	B-	I-	all classes
word-f1	84.83	88.44	91.60	82.39	95.78	97.59	89.07	90.29	94.97	97.13	93.17	94.25	83.50	86.48	90.59
phrase-f1	80.33		89.75		92.54		84.80		93.57		90.69		73.78		87.62

Table 1: results of our model on Peyma dataset. Two kinds of evaluation is used, namely word and phrase level. in word level evaluation B- assigns to first token of phrase and I- is for middle and last tokens.

	Event		Faculty		Location		Organization		Person		Product		
	B-	I-	B-	I-	B-	I-	B-	I-	B-	I-	B-	I-	all classes
word-f1	72.39	78.58	76.49	78.77	82.53	78.96	81.12	87.51	92.81	94.83	68.56	71.34	84.03
phrase-f1	58.45		69.53		80.73		78.01		91.46		62.97		79.93

Table 2: results of our model on Arman dataset.

	Team	Test Data 1								
		In Domain			Out Domain			Total		
		P	R	F1	P	R	F1	P	R	F1
1	MorphoBERT	88.7	85.5	87.1	86.3	83.8	85	87.3	84.5	85.9
2	Beheshti-NER-1	85.3	84.4	84.8	84.4	82.6	83.5	84.8	83.3	84
3	Team-3	87.4	77.2	82	87.4	73.4	79.8	87.4	75	80.7
4	ICTRC-NLPGroup	87.5	76	81.3	86.2	69.6	77	86.8	72.3	78.9
5	UT-NLP-IR	75.3	68.9	72	72.3	60.7	66	73.6	64.1	68.5
6	SpeechTrans	41.5	39.5	40.5	43.1	38.7	40.8	42.4	39	40.6
7	Baseline	32.2	45.8	37.8	32.8	39.1	35.7	32.5	41.9	36.6

Table 4: Phrase-level evaluation for subtask A: 3-classes

	Team	Test Data 1								
		In Domain			Out Domain			Total		
		P	R	F1	P	R	F1	P	R	F1
1	MorphoBERT	92.5	86.7	89.5	91.5	84	87.6	92.1	85.2	88.5
2	Beheshti-NER-1	90.5	87.2	88.8	89.7	85	87.3	90.1	85.8	87.9
3	Team-3	89.2	79.5	84.1	89.5	74.7	81.4	89.3	76.9	82.7
4	ICTRC-NLPGroup	90.1	78.2	83.7	88.7	70.2	78.4	89.4	73.5	80.7
5	UT-NLP-IR	87.3	71.9	78.9	86.4	61.1	71.6	86.9	65.7	74.8
6	SpeechTrans	66.8	38.3	48.7	66.2	35.2	46	66.6	36.4	47
7	Baseline	46.2	42.6	44.3	45.2	35.1	39.5	45.9	38.4	41.8

Table 5: Word-level evaluation for subtask A: 3-classes

results for subtask B is given in Table 6 and 7. we can achieve 83.5% and 88.4% F1 score respectively for phrase and word level evaluation.

	Team	Test Data 1								
		In Domain			Out Domain			Total		
		P	R	F1	P	R	F1	P	R	F1
1	MorphoBERT	88.4	84.8	86.6	86	83.1	84.5	87	83.8	85.4
2	Beheshti-NER-1	84.8	83.6	84.2	83.9	82	83	84.3	82.7	83.5
3	Team-3	87.4	77.3	82	87.3	72.8	79.4	87.3	74.7	80.5
4	ICTRC-NLPGroup	87	76.1	81.2	86.2	70.2	77.4	86.5	72.7	79
5	UT-NLP-IR	77.3	70.2	73.6	74.1	61.9	67.5	75.5	65.4	70.1
6	SpeechTrans	38	34.5	36.2	38.9	33.6	36	38.5	34	36.1
7	Baseline	32.8	45.7	38.2	32	38.1	34.8	32.4	41.3	36.3

Table 6: Phrase-level evaluation for subtask B: 7-classes

	Team	Test Data 1								
		In Domain			Out Domain			Total		
		P	R	F1	P	R	F1	P	R	F1
1	MorphoBERT	94	89.1	91.5	91.8	85.7	88.6	92.8	87.1	89.9
2	Beheshti-NER-1	91.4	87.3	89.3	89.7	85.7	87.7	90.4	86.5	88.4
3	Team-3	91.3	84.1	87.5	90.9	77.9	83.9	91.1	80.7	85.5
4	ICTRC-NLPGroup	89.2	83.1	86.1	89.8	76.5	82.6	89.7	79.4	84.2
5	UT-NLP-IR	92.7	79.3	85.4	91.1	68.4	78.1	91.9	73.1	81.4
6	SpeechTrans	76.1	32.9	45.9	74.9	30.3	43.2	75.7	31.5	44.5
7	Baseline	50.6	47.8	49.2	42.6	35.1	38.5	46.5	40.9	43.5

Table 7: Word-level evaluation for subtask B: 7-classes

details of evaluation for each class in subtask B is given in Table 7. as you see all teams have higher scores in Percent class and the worst score for many teams is for Time class.

	Team	Test Data 1							
		PER	ORG	LOC	DAT	TIM	MON	PCT	Total F1
1	MorphoBERT	90.4	80.3	87.1	78.9	71	93.6	96.8	85.4
2	Beheshti-NER-1	81.8	80.8	88	77.8	75.8	85.1	91.6	83.5
3	Team-3	79.9	77.2	83.9	74.7	64.3	92.1	97.4	80.5
4	ICTRC-NLPGroup	76.2	75.93	82.8	76	67.1	91.3	93.6	79
5	UT-NLP-IR	63.4	58.8	78.2	76.1	69.1	84.5	93.5	70.1
6	SpeechTrans	24.3	23.5	63.1	12	4.1	0.3	0.7	36.1
7	Baseline	23.5	38.1	44.2	41.6	30.3	13.7	36.6	36.3

Table 8: Details of phrase-level evaluation for subtask B: 7-classes

6 Conclucion

in this work we fine-tuned the pre-trained BERT model with a CRF layer in NER task for Persian language. our trained model achieved best results compared to the previous ones and ranked as the second team in NSURL competition. this work present BERT as a good transfer learning solution for solving low resource problems.

results show that our model could outperform previous methods with a dramatic difference. the reason for this could be using a big pre-trained model, BERT, which achieved state of the art results in English and proved to perform well with a less amount of data for training.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual String Embeddings for Sequence Labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Diego Mollá Aliod, Menno van Zaanen, and Daniel Smith. 2006. Named entity recognition for question answering. In *ALTA*.

Bogdan Babych and Anthony Hartley. 2003. Improving machine translation quality with automatic named entity recognition.

Alexei Baevski, Sergey Edunov, Yinhan Liu, Luke Zettlemoyer, and Michael Auli. 2019. Cloze-driven Pretraining of Self-attention Networks. *arXiv:1903.07785 [cs]*. ArXiv: 1903.07785.

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

Mohammad Hadi Bokaei and Maryam Mahmoudi. 2018. Improved Deep Persian Named Entity Recognition. In *2018 9th International Symposium on Telecommunications (IST)*, pages 381–386, Tehran, Iran. IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. 2009. Named entity recognition in query. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 267–274, New York, NY, USA. ACM.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

P. S. Mortazavi and M. Shamsfard. 2009. Named entity recognition in persian texts. 15th National CSI Computer Conference.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. ELMO: Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Hanieh Poostchi, Ehsan Zare Borzeshi, and Massimo Piccardi. BiLSTM-CRF for Persian Named-Entity Recognition. page 5.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. page 24.

Mahsa Sadat Shahshahani, Mahdi Mohseni, Azadeh Shakery, and Heshaam Faili. 2018. PEYMA: A Tagged Corpus for Persian Named Entities. *arXiv:1801.09936 [cs]*. ArXiv: 1801.09936.

Mehrnoush Shamsfard, Hoda Sadat Jafari, and Mahdi Ilbeygi. 2010. STeP-1: A set of fundamental tools for Persian text processing. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Nasrin Taghizadeh, Zeinab Borhani-fard, Melika Golestani-Pour, Mojgan Farhoodi, Maryam Mahmoudi, Masoumeh Azimzadeh, and Heshaam Faili. 2019. NSURL-2019 task 7: Named entity recognition (ner) in farsi. In *Proceedings of the first International Workshop on NLP Solutions for Under Resourced Languages*, NSURL '19, Trento, Italy.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *arXiv:1706.03762 [cs]*. ArXiv: 1706.03762.

Arabic Dialogue Act Recognition for Textual Chatbot Systems

Alaa Joukhadar
Higher Institute for Applied Sciences and Technology
Damascus, Syria
alaa.joukhadar@hiast.edu.sy

Huda Saghergy, Leen Kweider
IT Engineering Faculty,
Damascus University
huda.saghergy@gmail.com
leenkweider@gmail.com

Nada Ghneim
Higher Institute for Applied Sciences and Technology,
Damascus, Syria
nada.ghneim@hiast.edu.sy

Abstract

Automatic Dialogue Acts Recognition is considered a crucial step for semantic extraction in Natural Language Understanding and Dialogue Systems. In this paper, we introduce our work aiming to recognize the dialogue acts of the users in a Textual Dialogue system using Levantine Arabic dialect. Our Dialogue acts have 8 types: Greeting, Goodbye, Thanks, Confirm, Negate, Ask_repeat, Ask_for_alt, and Apology. Various Machine Learning algorithms -with different features have been used to detect the correct speech act categories: Logistic Regression, SVM, Multinomial NB, Extra Trees Classifier, Random Forest Classifier. We also used the Voting Ensemble method to make the best prediction from each classifier. We compared the results of the proposed models on a hand-crafted corpus in the restaurants orders and airline ticketing domain. The SVM algorithm with 2-gram has given the best results.

1 Introduction

Modeling and automatically identifying the structure of spontaneous dialogues is very important to better interpret and understand them. Speech act (or Dialogue act) recognition is considered an essential step in these models. Austin defines in (Austin, 1962) the speech act as the meaning of an utterance at the level of illocutionary force. In other words, the dialogue act is the function of a sentence (or its part) in the dialogue. For example, the function of a question is to request some information, while an answer shall provide this information.

The recognition of speech acts has gained considerable interest over the past two decades. Its significance derives from its broad range of applications such as: Tutorial Dialogue Systems (Ezen-Can and Boyer, 2014) (Rus et al., 2017), Machine Translation (Fukada et al., 1998), Animation of Talking Heads, Conversational Analysis (Fišel, 2007), Natural Speech Synthesis, Customer Service Conversation Outcomes Prediction (Oraby et al., 2017), etc.

Many researchers have proposed different approaches to recognize speech acts in different languages, such as English (Bothe et al., 2018) (Chen et al., 2018) (Elmadany et al., 2018), Korean (Kim et al., 2011) (Kim and Kim, 2018), German (Zarisheva and Scheffler, 2015), etc. They have developed different tag sets and corpora, investigating a variety of supervised (Tavafi et al., 2013) (Chen et al., 2018) (Kumar et al., 2018) and unsupervised machine learning techniques (Ezen-Can and Boyer, 2014) (Kristy Elizabeth Boyer, 2015) (Sherkawi et al., 2018).

The correct interpretation of the intents behind a speakers utterances plays a very essential role in determining the success of a dialogue. Therefore, the intents classification module lies at the very core of any dialogue system.

In general, chat bot systems can be composed of three basic components: Natural Language Understanding (NLU), Dialogue Manager (DM), and Natural Language Generation (NLG). The recognized dialogue acts (from the Natural Language Understanding component) are usually used as an input to the Dialogue Manager component, to help determine the next action of the system, such as giving correct information when the user is asking a question, and keeping quiet when the user is just acknowledging, or giving a simple comment. The Dialogue Acts taxonomy differs according to the

dialogue system domain.

The work presented here is part of a project that aims to build a domain-independent textual dialogue system in Levantine Arabic dialect. The concept of dialect in Arabic world is different from what is known in the west, as people do not use Standard Arabic in their day life but different dialects, which are very different from standard Arabic. Arabic dialects are generally classified by regions, such as in (Habash, 2010) where Arabic dialects were classified into North African, Levantine, and Egyptian. Our work considers the dialogues in Levantine (mostly Syrian) dialect.

The main contributions of this work are as follows:

- We provide an insight on the annotation of our Levantine Arabic Dialogue Act corpora used in restaurants orders and airline ticketing domain.
- We propose 5 learning models for Dialogue Act identification, along with different features.
- We evaluate and compare the accuracy of the different models on our Dialogue Act dataset.

Our paper is divided as follows: section 2 presents related works, section 3 is our proposed methods for the classification of the speech acts, including the proposed taxonomy and dataset. Section 4 presents the evaluation for our approach, and section 5 is the conclusion.

2 Related Works

Automatic recognition of dialogue acts is an important, yet still underestimated component of Human-Machine Interaction dialogue architecture. The research in this area have made great progress during the few last years.

In (Kumar et al., 2018) authors have built a hierarchical recurrent neural network using bidirectional LSTM as a base unit and the conditional random field (CRF) as the top layer to classify each utterance into its corresponding dialogue act. The hierarchical network learns representations at word, utterance, and conversation levels. The conversation level representations are input to the CRF layer, which takes into account all previous utterances and their dialogue acts. They validated their approach on Switchboard (SwDA) and Meeting Recorder Dialogue Act (MRDA) data sets, and show performance improvement over the state-of-the-art methods by 2.2% and 4.1% absolute points, respectively.

(Bothe et al., 2018) used simple RNN to model the context of preceding utterances. They used the domain independent pre-trained character language model to represent the utterances. Their proposed model was evaluated on the Switchboard Dialogue Act corpus and achieved an accuracy of 77.34% with context compared to 73.96% without context.

(Lee et al., 2016) have also presented a model based on recurrent neural networks and convolutional neural networks that incorporates the preceding short texts. They validated their model which achieved state-of-the-art results on three different datasets (DSTC 4, MRDA, and SwDA) for dialogue act prediction.

(Khanpour et al., 2016) have applied a deep LSTM structure to classify dialogue acts (DAs) in open-domain conversations (Khanpour et al., 2016). They found that the word embeddings parameters, dropout regularization, decay rate and number of layers have the greatest impact on the final system accuracy. They validated their model which outperformed the state-of-the-art on the Switchboard corpus by 3.11%, and MRDA by 2.2%.

In (Chen et al., 2018) authors proposed the CRF-Attentive Structured Network (CRF-ASN) to solve the problem in two steps. They first encoded the rich semantic representation on the utterance level by incorporating hierarchical granularity and memory enhanced inference mechanism. The learned utterance representation captured long term dependencies across the conversation. Next, they adopted the internal structured attention network to compute the dialogue act influence and specify structural dependencies in a soft manner. The approach enabled the soft-selection attention on the structural CRF dependencies and took account of the contextual influence on the nearing utterances. The method achieved better performance than several state-of-the-art solutions on SwDA and MRDA datasets.

(Wan et al., 2018) proposed an improved dynamic memory networks with hierarchical pyramidal utterance encoder. Moreover, they applied adversarial training to train the proposed model, which was evaluated on Switchboard dialogue act corpus and the MapTask corpus. Extensive ex-

periments showed that the model was robust and achieved better performance compared with some state-of-the-art baselines.

Concerning non English languages, some researches were focused on multilingual domain, such as the work of (Cerisara et al., 2018) who proposed a deep neural network approach that explores recurrent models to capture word sequences within sentences, and further studied the impact of pre-trained word embeddings. The model was validated on three languages: English, French and Czech, and the performance was consistent across these languages and comparable to the state-of-the-art results in English.

(Jahanbakhsh-Nagadeh et al., 2019) presented a dictionary-based statistical technique for Persian speech acts recognition. They used lexical, syntactic, semantic, and surface features to detect seven classes of speech acts. To evaluate their proposed technique, they implemented four classification methods including Random Forest, Support Vector Machine, Naive Bayes, and K-Nearest Neighbors. The experimental results demonstrated that the proposed method using RF and SVM had the best classification accuracy.

Arabic speech acts classification started to show few initiatives. Sherkawi et al. presented their rule-based model to detect Arabic Speech Act types (Sherkawi et al., 2017). The Expert System has been developed in a bootstrapping manner, to classify an utterance written in the Modern Standard Arabic (MSA) to one of the sixteen speech act types (Affirmation, Negation, Confirmation, Interrogation, Imperative, Forbidding, Wishing, Vocative, Prompting, Rebuke, Exclamation, Hope, Condition, Praise, Dispraise, Swear). The system was tested on a hand-crafted corpus of about 1500 MSA sentences.

In a following research, (Sherkawi et al., 2018) proposed a statistical based technique to recognize MSA Arabic speech acts. The proposed technique used surface features, cue words and contextual information. The authors compared the results of multiple machine learning algorithms (Decision Trees, Naïve Bayes, Neural Networks and SVM) on a corpus of 1500 MSA sentences. The Decision Tree algorithm had the best results.

(Elmadany et al., 2018) used the JANA corpus (4725 utterances in Egyptian Dialect) to create a statistical dialogue analysis model for recognizing utterances dialogue acts using a machine learning approach based on multi-classes hierarchical structure.

In (Graja et al., 2013), authors used the TuDiCoI corpus (12182 utterances in Tunisian Dialect) to develop a discriminative algorithm based on conditional random fields (CRF) to semantically label spoken Tunisian dialect turns which are not segmented into utterances.

(Shala et al., 2010) applied speech act classification for Arabic discourse using SVM, NB and Decision Trees machine learning classifiers on a dataset of about 400 MSA utterances collected from newspapers.

One more work on Arabic language was conducted by (Hijjawi et al., 2013) whose approach was based on Arabic function words (such as, هل do, كيف how) . They focused on questions/non-questions utterance classification using decision trees.

To the best of our knowledge, there are no studies on the Dialogue Act recognition of Levantine Arabic Dialect.

3 Our Approach

Our system is built to be domain independent, but in this work, we have applied it on both restaurants order and airline ticketing systems. Hereafter, we will introduce our taxonomy, our in-house built datasets, preprocessing steps, and the different machine learning algorithms used.

3.1 Our Taxonomy

Based on our chatbot system, we have adopted our own taxonomy of speech acts that are mostly used in restaurants orders and airline ticketing. We divided the utterances into 8 types: (Greeting, Goodbye, Thanks, Confirm, Negate, Ask_repeat, Ask_for_alt, and Apology).

Table 1 presents the descriptions of our taxonomy with corresponding examples.

3.2 Our Dataset

To our knowledge, there is no available corpus in the case of Levantine dialect that can be used to develop our dialogue system. Therefore, we manually built our own dataset, which consists of sentences from two domains: Restaurants Orders and Airplane Ticketing domain.

Our corpus contains a set of 873 sentences that were manually tagged. We started from scratch and collected the sentences from different sources:

- (63%) Obtained by means of crowdsourcing: We asked our colleagues to write sentences of how they would imagine a restaurant order or flight reservation conversation would go, then we manually tagged the sentences according to our taxonomy.
- (32%) Extracted from Levantine tweets related to the two domains: A python code was used to download tweets according to keywords for every class, these sentences were then manually labeled.
- (5%) A dataset collected in a previous food order chatbot project (Shbib et al., 2017).

Dialogue Act	%	Description	Example Utterance
Greeting	12.9	Greeting a person and saying hello.	مرحبا كيفك شو أخبارك؟ marHaba kyfak $w0 >xbarak Hello how are you what are you up to?
Goodbye	11.0	Ending a conversation or saying goodbye.	و عليكم السلام الله معك wA Ealaykum alsalam Al~A maEak Peace be upon you, goodbye
Thanks	13.0	Thanking a person.	شكراً كتير $ukran ktyr Thanks a lot.
Confirm	13.6	Confirming a yes/no question.	أي أكيد >y >akyd Yes of course
Negate	11.8	Negating a yes/no question.	لا ما بدي lA mA bid~i No I don't want it
Ask_repeat	12.7	Asking the speaker to repeat what he said.	ممكن تعيد شو قلت؟ Mumkin tEyd $w qlt Can you repeat what you said?
Ask_for_alt	12.6	Asking for alternative options if given.	شوفي عندك غير خيارات؟ $w fi Eandak gyr xayarat What other options do you have?
Apology	12.0	Apologizing to a person.	آسف lsif Sorry

Table 1: Our Dialogue Acts Taxonomy.

In another experiment, we have created a multi-labeled version of the dataset in order to apply multi-labeling classification techniques to the task. The dataset has been manually retagged such that each sentence can belong to one or more class (Dialogue act). For both experiments, the data was divided 80% for training and 20% for testing.

3.3 Preprocessing

Different steps were taken to preprocess the data. First, data was resampled to create equal number of sentences for each class.

No stop words were removed because stop words like (yes/ نعم , no/لا, Ok/ماشي...) are key features in the classification of speech acts.

We also tested the impact of using the stem of the words vs. the full form words, on the Dialogue acts classification. Therefore, we used the Arabic ISRI stemmer and compared the results using SVM classification algorithm.

3.4 Classification Algorithms

We used a set of different classifiers with different features and compared them. The classification algorithms that were tried were LogisticRegression (LR), Support Vector Machine (SVM), MultinomialNB (MNB), ExtraTreesClassifier (ET), and RandomForestClassifier (RF). We also used the voting ensemble method to make the best prediction from each classifier.

The features that were tried in this paper are TF-IDF, N-gram (N-grams were tried from 1 to 5), a combination of TF-IDF and N-gram. We also compared some feature selection methods such as Select From Model, Feature Union, and Recursive Feature Elimination (RFE). We implemented different experiments, and assessed their results using precision, recall and f-measure metrics.

The comparison results of the N-gram feature on Logistic Regression classifier is shown in Table 2. The table shows that 2-gram is the best feature with an accuracy of 0.89%.

Ngram	1	2	3	4	5
Accuracy	0.88	0.89	0.87	0.87	0.87

Table 2: Accuracy using Logistic Regression with N-gram (1-5)

In order to minimize the number of features in our model, and only select the best features, we compared some feature selection models and tested their results on our Logistic Regression classifier. Results are shown in Table 3.

Feature Selection Model	LR Accuracy
Select From Model (Extra Trees Classifier)	89%
Select From Model (Random Forest Classifier)	88%
Select From Model (Linear SVC)	88%
Select K-best (k = 800)	91%
Select Percentile (percentile = 50)	89%

Table 3: Results of different Feature Selection Models

The results presented in Table 3 show that Select k-best with k = 800 feature is the best feature selection model, thus it will be used in our next experiments.

4 Evaluation

In order to evaluate our approach, we implemented five machine learning models: LogisticRegression (LR), Support Vector Machine (SVM), MultinomialNB (MNB), ExtraTreesClassifier (ET), and RandomForestClassifier (RF).

We trained each classifier on different features and compared the results. The voting ensemble method was also evaluated for each feature. Table 4 compares the results obtained using our models.

	N-gram	TF-IDF	TF-IDF & N-gram
LR	**0.91**	0.89	0.86
SVM	0.89	0.86	0.85
RF	0.79	0.73	0.72
MNB	0.86	0.87	0.85
ET	0.88	0.87	0.86
Voting	0.90	0.89	0.87

Table 4: Results of different Machine Learning Models

The results show that Logistic Regression model using N-gram features outperforms the rest. Logistic Regression model improved the Dialogue Acts labeling accuracy over the SVM model by 2%.

To study the impact of using a stemmer in the preprocessing step, we used the ISRI stemming algorithm which is implemented for Modern Standard Arabic, and to our knowledge there is no stemmer for the Levantine Dialect. Results showed that using the MSA stemming did not improve the accuracy of the recognition. The MSA stemmer produces incorrect stems such as ابد/Abad for the word مابدي/mAbid y, and لمو/lamw for the word يسلمو/yislamw. These erroneous stems will be part of the features used, and will definitely affect the classification results.

In order to further analyze the results, we looked into the confusion matrix to know which labels are correctly/incorrectly assigned to sentences.

Figure 1 shows the confusion matrix of our Logistic Regression. We notice that the most errors were made in sentences that belong to the class "Thanks" and were predicted as "confirm".

	Confirm	Ask_Repeat	Goodbye	Apology	Negate	Ask_For_Alt	Greeting	Thanks
Confirm	23	0	0	0	**2**	**1**	0	**2**
Ask_Repeat	**2**	30	0	**2**	**3**	0	0	0
Goodbye	**1**	0	26	0	**2**	0	0	0
Apology	**2**	0	0	34	**1**	0	0	0
Negate	**1**	**1**	0	0	34	0	0	**1**
Ask_For_Alt	**1**	0	0	0	**1**	44	0	0
Greeting	**1**	0	1	0	0	0	38	0
Thanks	**4**	0	0	**1**	0	**1**	0	30

Figure 1: Confusion Matrix of the LR model

We noticed from the false predicted utterances that the sentences in fact belong to both classes, "Thanks" class and "Confirm" class. Table 5 shows some examples of these mislabeled sentences.

Dialog Act	Sentence
Thanks	أي ماشي يعطيك العافية بس لا تتأخرو بالتوصيل
Thanks	أي شكرا يعطيك العافية بس التوصيل اديش بيكلف
Thanks	أي يعطيك العافية

Table 5: Examples of "Thanks" sentences predicted as "Confirm"

To solve this problem, we re-labeled the data so each sentence would belong to more than one class, then we applied the One Vs. Rest multi-labeling classifier. The Results using different classifiers are shown in Table 6

Results show that our SVM classifier outperforms the rest of the classifiers with an accuracy of 86%.

One Vs. Rest Classifier	Accuracy
LR	0.84
SVM	**0.86**
RF	0.84
MNB	0.85
ET	0.82

Table 6: Results of the different Multi-labeling classifiers

5 Conclusion

In this paper, we have investigated different Dialogue act recognition models for Levantine Arabic language. The best model will be embedded into the Language Understanding component in our Arabic Conversational (Syrian Levantine Dialect) system.

We implemented different Machine Learning algorithms along with different features and feature selection methods. We evaluated the proposed techniques on a hand-crafted dataset in the restaurant's orders and airline ticketing domain. The best results were achieved using SVM model with 86% accuracy).

In the future, we intend to record a real restaurant and Ticket ordering conversations and create a new larger dataset, with real life situations and speech act sequences. This new dataset will allow us to take into consideration the whole context of the sentence in predicting the speech act of each utterance.

Building a Morphological Analyzer (or even a simple light stemmer) for Levantine (Syrian) Arabic, and using it in the preprocessing steps, will allow to extract many important features such as dialect negation tools (usually concatenated with the word itself, such as مارح/I will not, مابدي/I don't want), and this will improve the correct dialogue acts recognition.

References

John Langshaw Austin. 1962. *How to do things with words*. Harvard University Press, Cambridge.

Chandrakant Bothe, Cornelius Weber, Sven Magg, and Stefan Wermter. 2018. A context-based approach for dialogue act recognition using simple recurrent neural networks. Eleventh International Conference on Language Resources and Evaluation (LREC 2018).

Christophe Cerisara, Pavel Král, and Ladislav Lenc. 2018. On the effects of using word2vec representations in neural networks for dialogue act recognition.

Zheqian Chen, Rongqin Yang, Zhou Zhao Deng Cai, and Xiaofei He. 2018. Dialogue act recognition via crf-attentive structured network. pages 225–234. The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval ACM.

AbdelRahim Elmadany, Sherif Abdou, and Mervat Gheith. 2018. Improving dialogue act classification for spontaneous arabic speech and instant messages at utterance level. In *The 11th edition of the Language Resources and Evaluation Conference.*

Aysu Ezen-Can and Kristy Elizabeth Boyer. 2014. Combining task and dialogue streams in unsupervised dialogue act models. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 113–122.

Mark Fišel. 2007. Machine learning techniques in dialogue act recognition. pages 117–134.

Toshiaki Fukada, Detlef Koll, Alex Waibel, and Kouichi Tanigaki. 1998. Probabilistic dialogue act extraction for concept based multilingual translation systems. In *The Fifth International Conference on Spoken Language Processing*.

Marwa Graja, Maher Jaoua, and Lamia Hadrich Belguith. 2013. Discriminative framework for spoken tunisian dialect understanding. In *The International Conference on Statistical Language and Speech Processing*, pages 102–110. Springer.

Nizar Habash. 2010. *Introduction to Arabic natural language processing*. Synthesis Lectures on Human Language Technologies.

Mohammad Hijjawi, Zuhair Bandar, and Keeley Crockett. 2013. User's utterance classification using machine learning for arabic conversational agents. In *The 5th International Conference on Computer Science and Information Technology*, pages 223–232. IEEE.

Zoleikha Jahanbakhsh-Nagadeh, Mohammad-Reza Feizi-Derakhshi, and Arash Sharifi. 2019. A speech act classifier for persian texts and its application in identify speech act of rumors.

Hamed Khanpour, Nishitha Guntakandla, and Rodney D. Nielsen. 2016. Dialogue act classification in domain-independent conversations using a deep recurrent neural network. In *Khanpour, Hamed, Nishitha Guntakandla, Rodney D. Nielsen*. COLING.

Hark-Soo Kim, Choong-Nyoung Seon, and Jung-Yun Seo. 2011. Review of korean speech act classification: machine learning methods. 5:288–293.

Minkyoung Kim and Harksoo Kim. 2018. Dialogue act classification model based on deep neural networks for a natural language interface to databases in korean. In *IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 537–540. IEEE.

Aysu Ezen-Can Kristy Elizabeth Boyer. 2015. A tutorial dialogue system for real-time evaluation of unsupervised dialogue act classifiers: Exploring system outcomes. In *Proceedings of the international conference on artificial intelligence in education (AIED), Lecture Notes in Computer Science*, pages 105–114. Springer.

Harshit Kumar, Arvind Agarwal, Riddhiman Dasgupta, Sachindra Joshi, and Arun Kumar. 2018. Dialogue act sequence labeling using hierarchical encoder with crf. The Thirty-Second AAAI Conference on Artificial Intelligence.

Lee, Ji Young, and Franck Dernoncourt. 2016. Sequential short-text classification with recurrent and convolutional neural networks. In *NAACL*. arXiv preprint arXiv:1603.03827.

Shereen Oraby, Pritam Gundecha, Jalal Mahmud, Mansurul Bhuiyan, and Rama Akkiraju. 2017. How may i help you?: Modeling twitter customer serviceconversations using fine-grained dialogue acts. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, pages 343–355. ACM.

Vasile Rus, Nabin Maharjan, and Rajendra Banjade. 2017. Dialogue act classification in human-to-human tutorial dialogues. In *Innovations in Smart Learning*, pages 185–188, Singapore. Springer.

Lubna Shala, Vasile Rus, and Arthur C. Graesser. 2010. Automatic speech act classification in arabic. In *Subjetividad y Procesos Cognitivos Conference*, pages 284–292.

Boushra Shbib, Batool Ibo, Dima Qawoq, and Safa Alshaib. 2017. Arabic conersational agent for foor ordering.

Lina Sherkawi, Nada Ghneim, and Oumayma Al Dakkak. 2017. Arabic speech act recognition using bootstrapped rule based system.

Lina Sherkawi, Nada Ghneim, and Oumayma Al Dakkak. 2018. Arabic speech act recognition techniques.

Maryam Tavafi, Yashar Mehda, Shafiq Joty, Giuseppe Carenini, and Raymond Ng. 2013. Dialogue act recognition in synchronous and asynchronous conversations. In *Proceedings of the SIGDIAL 2013 Conference*, page 117–121.

Wan, Yao; Yan, Wenqiang, Jianwei, Gao; Zhao, Zhou ; Wu, Jian; S. Yu, and Philip. 2018. Improved dynamic memory network for dialogue act classification with adversarial training. In *The 2018 IEEE International Conference on Big Data (Big Data)*, pages 841–850. IEEE.

Elina Zarisheva and Tatjana Scheffler. 2015. Dialog act annotation for twitter conversations. In *Proceedings of the SIGDIAL 2015 Conference*, page 114–123.

Tha3aroon at NSURL-2019 Task 8: Semantic Question Similarity in Arabic

Ali Fadel, Ibraheem Tuffaha, and Mahmoud Al-Ayyoub

Jordan University of Science and Technology, Irbid, Jordan
{aliosm1997, bro.t.1996, malayyoub}@gmail.com

Abstract

In this paper, we describe our team's effort on the semantic text question similarity task of NSURL 2019. Our top performing system utilizes several innovative data augmentation techniques to enlarge the training data. Then, it takes ELMo pre-trained contextual embeddings of the data and feeds them into an ON-LSTM network with self-attention. This results in sequence representation vectors that are used to predict the relation between the question pairs. The model is ranked in the 1st place with 96.499 F1-score (same as the second place F1-score) and the 2nd place with 94.848 F1-score (differs by 1.076 F1-score from the first place) on the public and private leaderboards, respectively.

1 Introduction

Semantic Text Similarity (STS) problems are both real-life and challenging. For example, in the paraphrase identification task, STS is used to predict if one sentence is a paraphrase of the other or not (Madnani et al., 2012; He et al., 2015; AL-Smadi et al., 2017). Also, in answer sentence selection task, it is utilized to determine the relevance between question-answer pairs and rank the answers sentences from the most relevant to the least. This idea can also be applied to search engines in order to find documents relevant to a query (Yang et al., 2015; Tan et al., 2018; Yang et al., 2019).

A new task has been proposed by Mawdoo3[1] company with a new dataset provided by their data annotation team for Semantic Question Similarity (SQS) for the Arabic language (Schwab et al., 2017; Mahmoud et al., 2017; Alian and Awajan, 2018). SQS is a variant of STS, which aims to compare a pair of questions and determine whether they have the same meaning or not. The SQS in Arabic task is one of the shared tasks of the Workshop on NLP Solutions for Under Resourced Languages (NSURL 2019) and it consists of 12K questions pairs (Seelawi et al., 2019).

In this paper, we describe our team's efforts to tackle this task. After preprocessing the data, we use four data augmentation steps to enlarge the training data to about four times the size of the original training data. We then build a neural network model with four components. The model uses ELMo (which stands for Embeddings from Language Models) (Peters et al., 2018) pre-trained contextual embeddings as an input and builds sequence representation vectors that are used to predict the relation between the question pairs. The task is hosted on Kaggle[2] platform and our model is ranked in the first place with 96.499 F1-score (same as the second place F1-score) and in the second place with 94.848 F1-score (differs by 1.076 F1-score from the first place) on the public and private leaderboards, respectively.

The rest of this paper is organized as follows. In Section 2, we describe our methodology, including data preprocessing, data augmentation, and model structure, while in Section 3, we present our experimental results and discuss some insights from our model. Finally, the paper is concluded in Section 4.

2 Methodology

In this section, we present a detailed description of our model. We start by discussing the preprocessing steps we take before going into the details of the first novel aspect of our work, which is the

[1] https://www.mawdoo3.com

[2] https://www.kaggle.com

!	"	#	$	%	&	‘
(	)	+	*	,	-	.
/	:	;	<	=	>	?
@	[	\	]	^	_	`
{	|	}	~	،	؟	؛
’	‘	”	“			

Figure 1: Punctuation marks considered in the preprocessing step

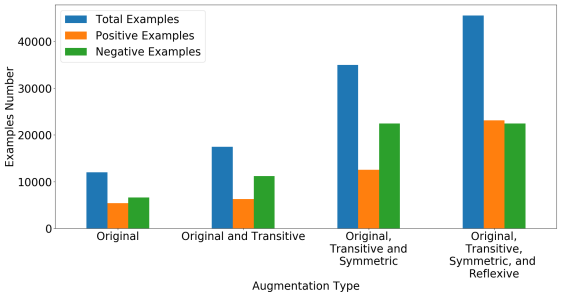


Figure 2: Number of examples per data augmentation step

data augmentation techniques. We then discuss the neural network model starting from the input all the way to the decision step. The implementation is available on a public repository.[3]

2.1 Data Preprocessing

In this work, we only consider one preprocessing step, which is to separate the punctuation marks shown in Figure 1 from the letters. For example, if the question was: “مرحبا، كيف الحال؟”, then it will be processed as follows: “مرحبا ، كيف الحال ؟”. This is done to preserve as much information as possible in the questions while keeping the words clear of punctuations.

2.2 Data Augmentation

The training data contains 11,997 question pairs: 5,397 labeled as 1 (i.e., similar) and 6,600 labeled as 0 (i.e., not similar). To obtain a larger dataset, we augment the data using the following rules.

Suppose we have questions A, B and C

- **Positive Transitive**:

  If A is similar to B, and B is similar to C, then A is similar to C.

- **Negative Transitive**:

  If A is similar to B, and B is *NOT* similar to C, then A is *NOT* similar to C.

  Note: The previous two rules generates 5,490 extra examples (bringing the total up to 17,487).

- **Symmetric**:

  If A is similar to B then B is similar to A, and if A is not similar to B then B is not similar to A.

  Note: This rule doubles the number of examples to 34,974 in total.

- **Reflexive**:

  By definition, a question A is similar to itself.

  Note: This rule generates 10,540 extra positive examples (45,514 total) which helps balancing the number of positive and negative examples.

After the augmentation process, the training data contains 45,514 examples (23,082 positive examples and 22,432 negative ones). Figure 2 shows the growth of the training dataset after each data augmentation step.

2.3 Model Structure

We now discuss our model structure, which is shown in Figure 3. As the figure shows, the model structure can be divided into the following components/layers: input layer, sequence representation extraction layer, merging layer and decision layer. The following subsections explain each layer/component in details.

2.3.1 Input

To build meaningful representations for the input sequences, we use the Arabic ELMo pre-trained model[4] to extract contextual words embeddings with size 1024 and feed them as input to our model. The representations extracted from the ELMo model are the averaged sum of word encoder and both first and second Long Short-Term Memory (LSTM) hidden layers. These representations are affected by the context in which they appear (Cheng et al., 2015; Peters et al., 2018; Smith, 2019). For example, the word “ذهب” will have different embedding vectors related to the following two sentences as they have different

[3] https://github.com/AliOsm/semantic-question-similarity

[4] https://github.com/HIT-SCIR/ELMoForManyLangs

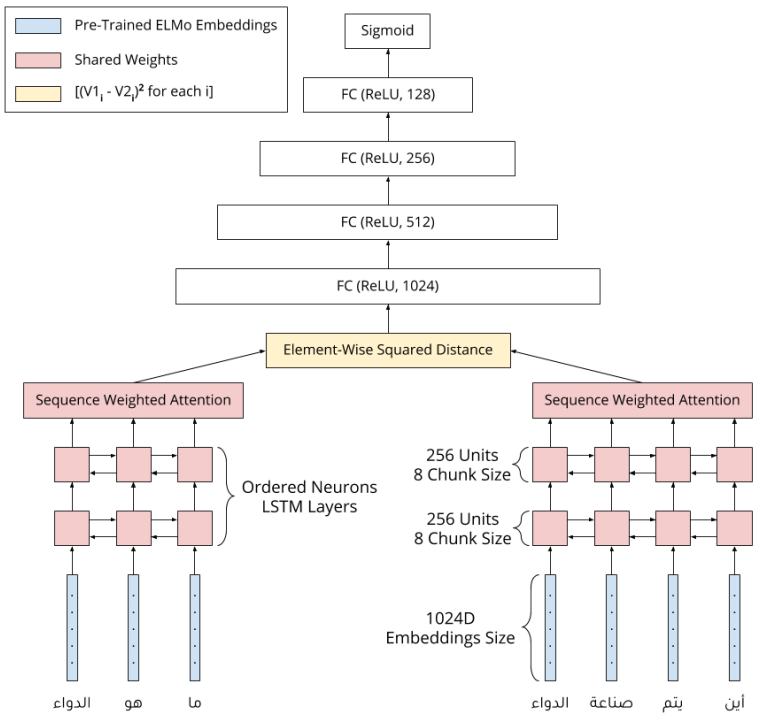


Figure 3: Model Structure

meanings ('gold' in the first sentence and 'went' in the second one):

ذهب علي كثير

Translation: Ali has a lot of gold.

ذهب علي بعيدا

Translation: Ali went away.

2.3.2 Sequence Representation Extractor

This component takes the ELMo embeddings related to each word in the question as an input and feeds them into two a special kind of bidirectional LSTM layers called Ordered Neurons LSTM (ON-LSTM)[5] introduced in (Shen et al., 2018) with 256 hidden units, 20% dropout rate, and 8 as the chunk size for each of them. Then, it applies sequence weighted attention[6] proposed by (Felbo et al., 2017) on the outputs of the second ON-LSTM layer to get the final question representation. This component uses the same weights to compute representations for each question in the pair. The details of this component are as follows (Shen et al., 2018).

Since NLP data are structured in a hierarchical manner, the authors of ON-LSTM (Shen et al., 2018) proposed a new form of update and activation functions (in order to enforce a bias towards structuring a hierarchy of the data) to the standard LSTM model reported below:

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \quad (1)$$
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \quad (2)$$
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \quad (3)$$
$$\hat{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (4)$$
$$h_t = o_t \circ \tanh(c_t) \quad (5)$$

The newly proposed activation function is $cumax = cumsum(softmax(x))$, where $cumsum$ denotes the cumulative sum function. Among the desired properties of this function is to control the updates on the memory cell such that the higher ranking neurons get updated less frequently (storing long-term and global information) compared to the lower ranking neurons, which are updated more frequently (storing short-term and local information). This makes the neurons updates dependent on each other in contrast to the updates on the standard LSTM neurons.

The following equations define the new master input and forget gates and the new memory cell update function based on the new activation func-

[5] https://github.com/CyberZHG/keras-ordered-neurons

[6] https://github.com/CyberZHG/keras-self-attention

tion:

$$\tilde{f}_t = cumax(W_{\tilde{f}} x_t + U_{\tilde{f}} h_{t-1} + b_{\tilde{f}}) \quad (6)$$

$$\tilde{i}_t = 1 - cumax(W_{\tilde{i}} x_t + U_{\tilde{i}} h_{t-1} + b_{\tilde{i}}) \quad (7)$$

$$w_t = \tilde{f}_t \circ \tilde{i}_t \quad (8)$$

$$\hat{f}_t = f_t \circ w_t + (\tilde{f}_t - w_t) \quad (9)$$

$$\hat{i}_t = i_t \circ w_t + (\tilde{i}_t - w_t) \quad (10)$$

$$c_t = \hat{f}_t \circ c_{t-1} + \hat{i}_t \circ \hat{c}_t \quad (11)$$

The attention mechanism (inspired by (Bahdanau et al., 2014; Yang et al., 2016)) allows the model to learn to decide the importance of each word and build the final question representation vector based on important words only, while tuning out less important words. With a single parameter, w_a, the attention mechanism can be described as follows:

$$e_t = h_t w_a \quad (12)$$

$$a_t = \frac{exp(e_t)}{\sum_{i=1}^{T} exp(e_i)} \quad (13)$$

$$v = \sum_{i=1}^{T} a_i h_i \quad (14)$$

The weight matrix w_a is the only new trainable parameter which learns the attention mechanism over the outputs of the second ON-LSTM layer. To calculate the importance scores, a_t, for each time step, it first multiplies each time step output, h_t, by the weight matrix, w_a, and normalizes the results using a Softmax function. Finally, the final sequence representation, v, is the weighted sum over all ON-LSTM outputs using the importance scores calculated earlier as weights.

2.3.3 Merging Technique

After extracting the representations related to each question, we merge them using pairwise squared distance function applied to the representation vectors of the two questions in each question pair. More formally, if $V1$ and $V2$ are these representation vectors, then, the merged representation vector Vm can be expressed as follows:

$$Vm = \begin{bmatrix} (V1_1 - V2_1)^2 \\ (V1_2 - V2_2)^2 \\ \vdots \\ (V1_{512} - V2_{512})^2 \end{bmatrix} \quad (15)$$

This component allows for the Symmetric augmentation step (Section 2.2) to enhance the results, since the (A, B) examples are computationally different (in the back propagation step) from the (B, A) examples.

2.3.4 Deep Neural Network

The final component is a deep neural network that consists of four fully-connected layers with 1024, 512, 256, and 128 units using ReLU activation function and 20% dropout rate applied to each layer. This network takes the merged representation vector, Vm, as an input and predicts the label using a Sigmoid function as an output.

3 Experiments and Results

In this section, we start by discussing our experimental setup. We then discuss all experiments conducted and provide detailed analysis of their results.

3.1 Experimental Setup

All experiments discussed in this work have been done on the Google Colab[7] (Carneiro et al., 2018) environment using Tesla T4 GPU accelerator with the following hyperparameters:

- Optimizer: Adam
- Learning Rate: 0.001
- Loss Function: Binary Cross Entropy
- Batch Size: 256
- Number of Epochs: 100

The experiments are divided into two sets. The first set aims to explore the effect of the Recurrent Neural Network (RNN) cell type, while the second set aims to explore the effect of the data augmentation techniques mentioned in Section 2.2.

For each experiment, five models are trained and the following results are reported:

- Minimum F1 score gained on the test set.
- Maximum F1 score gained on the test set.
- Average F1 score gained from the five trained models.
- Majority Voting F1 score gained by ensembling the five trained models.

Table 1: Model size and training time for each RNN cell type

RNN Cell	#Params	Training Time
GRU	4,363K	55.2s/epoch - 1.53 hours
LSTM	5,413K	58.2s/epoch - 1.61 hours
ON-LSTM (Chunk: 4)	5,938K	74.2s/epoch - 2.06 hours
ON-LSTM (Chunk: 8)	5,675K	74.4s/epoch - 2.06 hours

Table 2: Model F1-score using different RNN cell types

Leaderboard	RNN Cell	Min	Max	Avg	Vote
Public	GRU	94.075	94.793	94.613	95.242
	LSTM	94.614	95.152	94.901	95.062
	ON-LSTM (Chunk: 4)	94.524	95.601	95.242	96.140
	ON-LSTM (Chunk: 8)	**95.601**	**95.780**	**95.691**	**96.499**
Private	GRU	93.271	94.194	93.855	94.579
	LSTM	93.925	94.271	94.040	94.117
	ON-LSTM (Chunk: 4)	93.810	94.425	94.224	94.732
	ON-LSTM (Chunk: 8)	**94.002**	**94.463**	**94.309**	**94.848**

3.2 Effect of RNN Cell Type

In this experiments set, we use the same structure described in Section 2.3 while changing the RNN cell type only. We use all 45,514 examples from the augmented dataset in the training process. The tested RNN cells are: Gated Recurrent Unit (GRU) (Cho et al., 2014), LSTM (Hochreiter and Schmidhuber, 1997) and ON-LSTM (Shen et al., 2018). The latter one is tested using two chunk sizes, 4 and 8, in order to explore the effect of chunk size on the training process and the size of the model. Table 1 shows the model size in terms of trainable parameters and the training time for each RNN cell type, while Table 2 shows the F1-scores of the model using different RNN cells. Best results are shown in bold. The tables show that while GRU cells are the most efficient, the ON-LSTM cells (with chunk size 8) are the most effective (in terms of all considered measures).

3.3 Effect of Data Augmentation

In this experiments set, we use the RNN cell type that gives the best results in Section 3.2 (ON-LSTM with chunk size 8) and the same model structure described in Section 2.3 to explore the effect of data augmentation steps mentioned in Section 2.2.

The data augmentation steps have an effect on two factors, the training time and the accuracy measurement (F1-score). Table 3 shows the average training time over five runs for each data augmentation step. Moreover, Table 4 shows the F1-scores of the trained model using different data augmentation types, best results shown in bold.

The tables show that each augmentation step affects the model's efficiency negatively. This is expected since each step incrementally increases the size of the dataset. On the other hand, not each increment step has a positive effect on the model's effectiveness. Such trends are worth exploring in a more exhaustive study. Finally, it is worth mentioning that the last experiments in both experiment sets are the same. So, they both have the same results.

3.4 Other Attempts

We test several other techniques to explore how they might affect our model. For example, using pre-trained FastText (Bojanowski et al., 2017) embeddings as an input to our model yields worse F1-score on both public and private leaderboards with 94.254 and 93.118, respectively, compared with the ELMo contextual embeddings model. In another experiment, we use the thought vector outputted from the second ON-LSTM layer as input for the decision component. However, the sequence weighted attention gives better results by about 1 point of the F1-score. Moreover, an attempt to overcome the weakness of the Arabic ELMo model is done by translating the data to

[7] https://colab.research.google.com

Table 3: Model training time for each data augmentation step: O, T, S, and R, which stand for Original, Transitive, Symmetric, and Reflexive, respectively

Data Augmentation	Examples Number	Training Time
O	11,997	20.0s/epoch - 0.55 hours
O+T	17,487	29.4s/epoch - 0.81 hours
O+T+S	34,974	57.0s/epoch - 1.58 hours
O+T+S+R	45,514	74.4s/epoch - 2.06 hours

Table 4: Model F1-score using different data augmentation types: O, T, S, and R, which stand for Original, Transitive, Symmetric, and Reflexive respectively

Leaderboard	Data Aug.	Min	Max	Avg	Vote
Public	O	93.626	94.703	94.200	94.973
	O+T	93.177	94.434	93.877	94.793
	O+T+S	94.344	94.793	94.631	95.421
	O+T+S+R	**95.601**	**95.780**	**95.691**	**96.499**
Private	O	93.425	93.810	93.632	94.655
	O+T	92.464	93.771	93.232	94.156
	O+T+S	93.579	94.002	93.763	94.655
	O+T+S+R	**94.002**	**94.463**	**94.309**	**94.848**

Figure 4: Representations extracted from sequence weighted attention layer for questions of the form: How to prepare ‘something’?

Figure 5: Representations extracted from sequence weighted attention layer for questions of the form: What is the definition of ‘something’?

English using Google Translate[8] and treating the problem as an English SQS problem instead, but the results are much worse with 88.868 and 87.504 F1-scores on public and private leaderboards, respectively. This is probably because a lot of information is lost during the translation process.

3.5 Discussion

This section briefly analyzes the questions representations learnt by our model. With the sequence weighted attention layer, the model reduces all the information about the sequence extracted using the ON-LSTMs down to a 512 fixed-size vector. By extracting these vectors from our best model and plotting them on a 2D plane using t-SNE (Maaten and Hinton, 2008) dimensionality reduction algorithm, we notice some very useful observations. For example, the model learns to map questions that ask about the same thing to have nearby representations in the vector space such as the questions in Figure 4 with the form: “How to prepare ‘something’?”. The same thing goes for the questions in Figure 5 with the form: “What is the definition of ‘something’?”. In a similar manner, in Figure 6, the questions ask about different types of languages like “What is the formal language in Portugal?” and “What is PHP language?” are close, as well as, the questions in Figure 7 that ask about places like “Where is Sweden?”, “Where is the Karak area in Jordan?”, and “Where is the Kremlin Castle?”.

To further illustrate the usefulness of the sequence weighted attention layer, Figure 8 shows that the attention layer learns to focus more on the key words in the questions that would determine what the question is actually asking about. This allows the model to make better decisions for whether the the questions are similar or not, even if the questions have similar words but ask about different things. The first and second questions

[8] https://translate.google.com

Figure 6: Representations extracted from sequence weighted attention layer for questions that ask about different language types

Figure 7: Representations extracted from sequence weighted attention layer for questions that ask about different places

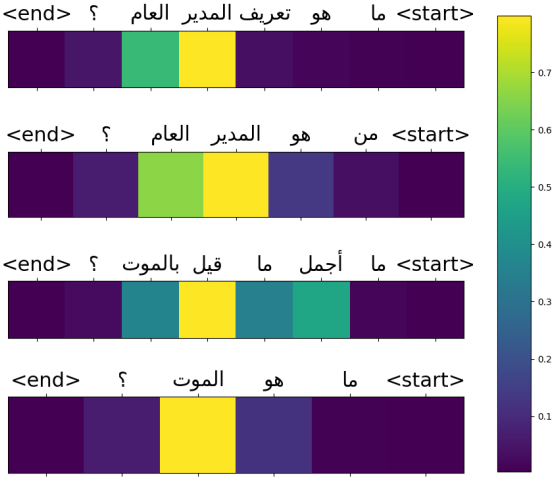


Figure 8: Weights per word from sequence weighted attention layer on four different examples

in Figure 8 ask about "What is the general manager?". So, the attention layer focuses on "the general manager" which is "المدير العام". However, in the third and fourth questions, one asks "What is the most beautiful thing that is said about death?" and the other ones asks "What is death?", although both questions are related to "death" which is "الموت" but the attention layer distinguishes them as not similar, where in the former one, the focus is concentrated by order on the words "قيل", "أجمل" and "بالموت" ("said", "most beautiful" and "death"), while the latter one focuses mostly on "الموت" ("death").

4 Conclusion

In this paper, we described our team's effort on the semantic text question similarity task of NSURL 2019. Our top performing system utilizes several innovative data augmentation techniques to enlarge the training data. Then, it takes ELMo pre-trained contextual embeddings as an input and builds sequence representation vectors that are used to predict the relation between the question pairs. The model was ranked in the 1st place with 96.499 F1-score (same as the second place F1-score) and the 2nd place with 94.848 F1-score (differs by 1.076 F1-score from the first place) on the public and private leaderboards, respectively.

Acknowledgments

We gratefully acknowledge the support of the Deanship of Research at the Jordan University of Science and Technology for supporting this work via Grant #20180193.

References

Mohammad AL-Smadi, Zain Jaradat, Mahmoud Al-Ayyoub, and Yaser Jararweh. 2017. Paraphrase identification and semantic text similarity analysis in arabic news tweets using lexical, syntactic, and semantic features. *Information Processing & Management*, 53(3):640–652.

Marwah Alian and Arafat Awajan. 2018. Arabic semantic similarity approaches-review. In *2018 International Arab Conference on Information Technology (ACIT)*, pages 1–6. IEEE.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Tiago Carneiro, Raul Victor Medeiros Da Nóbrega, Thiago Nepomuceno, Gui-Bin Bian, Victor Hugo C De Albuquerque, and Pedro Pedrosa Reboucas Filho. 2018. Performance analysis of google colaboratory as a tool for accelerating deep learning applications. *IEEE Access*, 6:61677–61685.

Jianpeng Cheng, Zhongyuan Wang, Ji-Rong Wen, Jun Yan, and Zheng Chen. 2015. Contextual text understanding in distributional semantic space. In *Pro-*

ceedings of the 24th ACM International on Conference on Information and Knowledge Management, pages 133–142. ACM.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *arXiv preprint arXiv:1708.00524*.

Hua He, Kevin Gimpel, and Jimmy Lin. 2015. Multi-perspective sentence similarity modeling with convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1576–1586.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 182–190. Association for Computational Linguistics.

Adnen Mahmoud, Ahmed Zrigui, and Mounir Zrigui. 2017. A text semantic similarity approach for arabic paraphrase detection. In *International Conference on Computational Linguistics and Intelligent Text Processing*, pages 338–349. Springer.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Didier Schwab et al. 2017. Semantic similarity of arabic sentences with word embeddings.

Haitham Seelawi, Ahmad Mustafa, Hesham Al-Bataineh, Wael Farhan, and Hussein T. Al-Natsheh. 2019. NSURL-2019 task 8: Semantic question similarity in arabic. In *Proceedings of the first International Workshop on NLP Solutions for Under Resourced Languages*, NSURL '19, Trento, Italy.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2018. Ordered neurons: Integrating tree structures into recurrent neural networks. *arXiv preprint arXiv:1810.09536*.

Noah A Smith. 2019. Contextual word representations: A contextual introduction. *arXiv preprint arXiv:1902.06006*.

Chuanqi Tan, Furu Wei, Wenhui Wang, Weifeng Lv, and Ming Zhou. 2018. Multiway attention networks for modeling sentence pairs. In *IJCAI*, pages 4411–4417.

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.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Motivations, Challenges, and Perspectives for the Development of a Deep Learning based Automatic Speech Recognition System for the Under-resourced Ngiemboon Language

Patrice A. Yemmene

School of Engineering
University of Saint Thomas, MN, USA
yemm2299@stthomas.edu

Laurent Besacier

Laboratoire Informatique de Grenoble
University of Grenoble, France
laurent.besacier@univ-grenoble-alpes.fr

Abstract

Nowadays, a broad range of speech recognition technologies (such as Apple Siri and Amazon Alexa) are developed as the user interface has become ever convenient and prevalent. Machine learning algorithms are yielding better training results to support these developments in Automatic Speech Recognition (ASR). However, most of these developments have been in languages with worldwide, political, economic and/or scientific influence such as English, Japanese, German, French, and Spanish, just to name a few. On the other hand, there has been little or no development of ASR systems (or language technologies) in most minority and under-resourced languages of the world, especially those spoken in Sub-Sahara Africa. One of such languages is the Ngiemboon language which is the focus of this paper. The Ngiemboon language is a Grassfield Bantu language spoken in the West Region of Cameroon (Africa) by about 400,000 people. This paper highlights the motivations, challenges and perspectives inherent in a work in progress (speech data collection is underway) to build a Deep Learning based Automatic Speech Recognition System for this minority under-resourced Cameroonian local language. This paper introduces the issues critical to conducting research in Speech Processing in this language

1. Introduction

Automatic Speech Recognition is "the process and the related technology for converting the speech signal into its corresponding sequence of words or other linguistic entities by means of algorithms implemented in a device, a computer, or computer clusters" (Li and O'Shaughnessy, 2003). As an active field of research, Automatic Speech Recognition has told significant stories for a few decades. "Early attempts to design systems for automatic speech recognition were mostly guided by the theory of acoustic-phonetics, which describes the phonetic elements of speech (the basic sounds of the language) and tries to explain how they are acoustically realized in a spoken utterance" (Juang and Rabiner, 2005). These efforts date back to the early 50s. Since then, ASR has yielded incredible development in a broad range of commercial technologies where Speech Recognition as the user interface has become ever useful and pervasive.

However, most of these developments have been in languages with strong scientific, political, and/or economic influences such as English, German, French, and to some extent Japanese and Spanish, just to name a few. Historically, most of these languages have always enjoyed social prestige and their extensive vocabulary has given them prominence in the world of commerce. It is worth noting that ASR research and innovation in these languages are significant and continuous. On the contrary, there has been little or no research and development efforts in ASR and other Human Language Technologies in most minority languages of the world, particularly those spoken in Sub-Sahara Africa. Yet, these languages serve as the main vector for the socio-economic development of communities where they are spoken. In this paper, we highlight the motivations, challenges, and perspectives that must be considered in building Human Language Technologies, more precisely an Automatic Speech Recognition System for the Ngiemboon language.

1.1 Paper objective and contribution

A surge of interest in the development of technologies in African languages is emerging. The African Languages in the Field: speech Fundamentals and Automation (ALFFA)1 project (spearheaded in France by the "Laboratoire Informatique de Grenoble"

[1] http://alffa.imag.fr/

of the Grenoble Alpes University) is a great example and has been leading significant efforts in the automation of languages spoken in sub-Sahara Africa. Researchers interested in African Languages hope to contribute to the history of Language Technologies innovations as it is being written across the continent. The objective of this paper is to contribute to the research on the development of Language Technologies in African Languages. As a pioneer research project on ASR for the Ngiemboon language, this work will provide a guide for work in Natural Language Processing (NLP) in minority and under-resourced language in Cameroon and other Sub-Sahara African languages.

1.2 The choice of the language for ASR

The authors of this paper both share a very strong interest in the automation of minority and under-resourced African languages. In fact, under different circumstances, each of them carried out research on some of these languages and have become aware of the challenges faced when working on the digitization and automation of minority languages of Africa. One of such challenges is "to bridge the gap between language experts (the speakers themselves) and technology experts (system developers). Indeed, it is often almost impossible to find native speakers with the necessary technical skills to develop ASR systems in their native language" (Besacier et al. 2014). It becomes obvious that a degree of collaboration between native speakers and systems developers is essential to addressing this identified challenge. Fortunately, one of the authors of this paper is a native speaker of the Ngiemboon language and a trained linguist who has contributed to the development of an already published trilingual – French – English – Ngiemboon dictionary. The availability of a native speaker explains the authors' preference in exploring ASR for the Ngiemboon language.

2. Motivations

The rationale for ASR research and development in under-resourced, minority languages spoken in Sub-Sahara Africa such as the Ngiemboon language is grounded in a unique sociolinguistic context, an observation of existing literacy gap, a recognition of advances in technology, a paradigm shift in human rights priorities and scientific discoveries as well as an understanding of the implications of these for economic and community development. In this section, we highlight these motivation factors.

2.1 Sociolinguistic considerations

The nation of Cameroon is home to about 247 local languages, two official languages (French and English) and Pidgin English (Echu, 2004). In their linguistic choices, it is estimated that 73% of Cameroonians use their mother tongue (a local Cameroonian language) instead of a foreign language (English and/or French), despite the peaceful coexistence of these Indo-European languages with Cameroonian languages. This linguistic choice is explained by the fact that Cameroonian local languages are spoken either in the village of their native speakers, their homes, and often used for heritage and cultural identification (Ngefac, 2010). In this diverse linguistic landscape, many industries and fields currently access ASR only in the high-resourced languages of French and English where "presently ASR systems find a wide variety of applications in the following domains; Medical Assistance; Industrial Robotics; Forensic and Law enforcement; Defense & Aviation, Telecommunications Industry; Home Automation and security Access Control; I.T. and Consumer Electronics" (Vajpai and Bora, 2016). As vital as these might be, they are still a luxury for speakers of Cameroonian local languages, including Ngiemboon speakers. Speakers of Ngiemboon, as well as speakers of other Cameroonian languages, prefer the use of their mother tongue in daily communication (Echu, 2004). What if vital ASR applications were developed in the Ngiemboon language as well? It would be an opportunity with great excitement for Ngiemboon speakers' economic, social, and community development.

2.2 Literacy gap

Over two decades ago, analysis of the literacy landscape in Cameroon reported that "four million Cameroonians above fifteen years of age are illiterate. This includes people who never went to school and those who have lapsed back into illiteracy. The Cameroon population is about eleven million people. This is a young population. About 60 percent of Cameroonians are below twenty-five years old. The accuracy of literacy rate estimates is doubtful and could be higher" (Tadadjeu, 2004). It is highly likely that the population of Cameroon has grown significantly since then. A 2018 US Federal Government civilian foreign intelligence service report suggested that about 25 million heads were counted in Cameroon with a 75 % literacy rate. This estimate assumes that about 6 million individuals or more living in Cameroon were illiterate as early as last year. We do not have any reason to believe this has changed much during the last few months. In an illiteracy context such as this one, the use of oral communication is

preponderate and convenient. Human-Machine interaction via voice has great potential for economic and community development.

2.3 Economic and community development motivations

In recent years, the mobile telephone industry has experienced a significant boom, in this part of the world where cell phone usage has become very pervasive. For example, "the number of subscribers has risen tremendously; in 2012, there were approximately nine million telephone users in Cameroon, a country with a population of twenty million inhabitants. …. These numbers are certainly below the average" (Moraa, 2012). It is believed that these numbers have changed significantly leading to increased opportunities for Human-Machine interaction.

Because mobile phones only require basic literacy, they are accessible to a large segment of the population regardless of their literacy status. In addition to voice communication, they allow for the transfer of data, which can be used in the context of speech applications for the purposes of health, education, commerce and/or governance. Mobile phones can be used as a mechanism to ensure greater participation of different segments of the population in community development efforts. Innovations along these lines will increase the likelihood of connecting Ngiemboon speakers to vital information that they need to enhance the quality of their lives and contribute to the development of their communities.

2.4 Legal motivations

Another motivation to develop ASR in Ngiemboon language is to allow this linguistic community the opportunity to exercise one of their fundamental rights expressed in article 40 of the Universal Declaration of Linguistic Rights, "In the field of information technology, all language communities are entitled to have at their disposal equipment adapted to their linguistic system and tools and products in their language, so as to derive full advantage from the potential offered by such technologies for self-expression, education, communication, publication, translation and information processing and the dissemination of culture in general". Furthermore, in article 47, "All members of a language community are entitled to have at their disposal, in their own language, all the means necessary for the performance of their professional activities, such as documents and works of reference, instructions, forms, and computer equipment, tools and products". The identified linguistic right aligns with 21st-century human rights priority. In addition to giving this linguistic community the opportunity to exercise one of its fundamental rights, it is also a fascinating endeavor to develop an ASR system for the Ngiemboon language.

2.5 Scientific motivations

The Development of a Speech Recognition system in the Ngiemboon language will play a great role in the revitalization and safeguarding of the language. It will also provide a framework for the digital documentation of the Ngiemboon language. Given the linguistic complexities and peculiarities of the Ngiemboon language (as we will be highlighting in the next section) an ASR research in Ngiemboon may yield discoveries that could add to existing and growing scientific knowledge in this exciting and challenging area of ASR in under-resourced languages.

3. A brief overview of the Ngiemboon language

3.1 Sociolinguistic overview

The Ngiemboon language is part of the Bamileke subgroup of the Eastern Grassfields language family, spoken in the West Region of Cameroon (Anderson, 2008). It has an estimated number of 400,000 speakers. Using EGIDS (Expanded Graded Intergenerational Disruption Scale), a tool used to measure the status of a language in terms of endangerment or development, Ethnologue estimates that the Ngiemboon language is developing. In other words, the language is in vigorous use, with literature in a standardized form being used by some, though this is not yet widespread or sustainable. The language has 5 dialectal variations (Batcham, Balessing, Bangang, Bamougong, Balatchi). From personal observations, lexico-statistic variations among these dialects are very minimal, and mutual intelligibility substantially high.

3.2 Linguistic overview

The Ngiemboon language has very complex linguistic characteristics. "Roots consist of the following C(S)V(C)(V)., ie an obligatory root-initial consonant, and optional semi-vowel, an obligatory vowel, and optional final consonants and vowel" (Anderson, 2008). Anderson (2008) further describes this as "an obligatory root-initial consonant, an optional semivowel, and obligatory vowel, an optional consonant, and an optional final vowel". Its nasal prefixes are syllabic. The language has 16 underlying consonants.

		Labials	Corons	Velars
Stops:	Voiceless		τ	κ
	Voiced	β	δ	γ
Affricates:		πϕ	τσ	
Fricatives:	Voiceless	ϕ	σ	
	Voiced	ϖ	ζ	
Nasals:		m	n	ñ
Semivowels:			j	w

Ngiemboon underlying consonants proposed by Anderson, 2008

These consonants may exhibit variations based on either their position in the root, the vowel that precedes or follows them, resulting in the following phonetic consonant chart.

Ngiemboon phonetic consonants proposed by Anderson, 2008

		Bi	LD	Den	Alv	Ret	A-P	Vel	Uvl	Glo
Stops:	Voiceless	p		t̪	t	ʈ		k	q	ʔ
	Voiced	b		d̪		ɖ		g		
	Unreleased	p̚			t̚				q̚	ʔ̚
Affricates: Voiceless			p͡f		ts		t͡ʃ			
	Voiced		b͡v		dz		d͡ʒ			
Fricatives:	Voiceless		f		s		ʃ	x		
	Voiced	β	v		z		ʒ	ɣ	ʁ	
Nasals:		m	ɱ	n̪	n	ɳ	ɲ	ŋ		
	Unreleased	m̚						ŋ̚		
Liquids:				l̪	r	ɭ				
Semivowels:	Unrounded						j	ɯ		
	Rounded						ɥ	w		

Additionally, the Ngiemboon language has a few underlying vowels, a total of seven identified by Anderson (2008). Most of these vowels can be modified by length and/or nasalization. The following chart exhibits the underlying vowels, long and short oral vowels, as well as short and long nasalized vowels in this grammatically rich tonal Bantu language.

Ngiemboon underlying vowels proposed by Anderson, 2008.

This calls for a very complex vowel system. The chart below highlights an overview of this vowel system:

Underlying Vowels	Short oral	Long oral	Short nasalized	Long nasalized
/i/	i	iː	ĩ	ĩː
/e/	e	eː	ẽ	ẽː
/ɛ/	ɛ	ɛː		
/a/	a	aː	ã	
/ɔ/	ɔ	ɔː		ɔ̃ː
/o/	o	oː	õ	õː
/u/	u	uː	ũ	ũː

We should also "recognize the four phonetic semivowels [that enrich the Ngiemboon phonology] as underlying units, even though the parallel four high vowels in Ngiemboon are not all underlying (Anderson, 2008). More details on Ngiemboon phonology can be found in Anderson, 1976a.

The complexity of the Ngiemboon language is extended to its tonal system (Anderson, 2008). In other words, the Ngiemboon language is a tonal language. A tonal language is a language that has "morphemes whose surface pitch (acoustically understood as the fundamental frequency with which corresponds articulatory the rate at which the vocal cords vibrate at any point in time) patterns contrast with each other in one or more comparable environment" (Snider, 2017).

This tonal system has four main tone melodies on noun stems. "For example, monosyllabic noun stems with a preceding low-tone prefix display the following stem tones in isolation: Rising, Down stepped High, Low, and Low-falling" (Anderson, 2008). Tone perturbations are considered the most complex part of this tonal system, with tones of individual words changing when these are put into sentences, or tone changing in conjugated verbs.

3.3 An under-resourced language

An under-resourced language can be defined as "a language with some (if not all) of the following aspects: lack of a unique writing system or stable orthography, limited presence on the web, lack of linguistic expertise, lack of electronic resources for speech and language processing such as monolingual corpora, bilingual electronic dictionaries, transcribed speech data, pronunciation dictionaries, vocabulary lists, etc." (Berment, 2004). In addition to providing this definition, a quantitative approach that can be used to determine the level of computation/automation of a language is suggested. He assigns a level of criticality, Cᴋ, to a service or resource available in the language, a grade Nᴋ, and an average which he calls index-σ. Berment suggests that the weighted average for less-resourced languages should be between 0 – 9.99, the weighted average for resourced languages between 10 – 13.99, and the weighted average for highly resourced languages between 14 – 20. When these criteria are applied to the Ngiemboon language, the results are as follow:

	Services/Resources	Criticality Ck (0 to 10)	Grade Nk (/20)	weighted average (Criticality * grade)
Word processing	Text entry	0	0	0
	Visualization/ printing	0	0	0
	Find/replace	0	0	0
	Text selection	0	0	0
	Sorting	0	0	0
	Lexical Spell check	0	0	0
	Grammatical spell check	0	0	0
	Style	0	0	0
Speech processing	Voice Synthesis	0	0	0
	Speech Recognition	0	0	0

Translation	Automatic translation	0	0	0
OCR	Optical Character Recognition	0	0	0
Resources	Bilingual Dictionary	8	0	8
	Monolingual Dictionary	0	0	0
Total		**8**		8
Average				**8/8 = 1**

The Ngiemboon language is therefore an under-resourced language, based on this approach.

3.4 A minority language

The definition of a minority language is quite complex. However, for this article, we would consider the definition of the European Charter for Regional or Minority Languages which states that " minority languages mean languages that are:

- traditionally used within a given territory of a State by nationals of that State who form a group numerically smaller than the rest of the State's population; and
- different from the official language(s) of that State"

Though the Ngiemboon language is not a European Language, we believe this definition fits it well, because it is spoken by about 400,000 people in a country that claims about 25 million inhabitants today. According to Ngefac (2010), there are over 247 languages spoken in Cameroon. A review of the Cameroon constitution implies that these languages can be classified into two broad categories: national languages and official languages (English and French). Both national and official languages coexist. The Ngiemboon language is one of the many national languages. We may also refer to it in this paper as a Cameroonian local language.

The understanding that the Ngiemboon language is a minority language can be further justified by the fact that though it has a writing system, it is not certain whether this writing system is complete. Although efforts have been made to develop the language in its written form, it still lacks an in-depth written grammatical description. In addition to this, written literature is very limited and is narrowed to Christian literature. Its presence on the web is very insignificant, and the language is spoken mostly in its geo-linguistic area, where it enjoys a lesser social prestige, compared to the French language spoken in the same geo-linguistic area as well as the whole country and beyond.

4. Challenges in developing an ASR system for the Ngiemboon language

The previous section clearly shows that the Ngiemboon language is a minority low-resourced language. There are challenges related to the development of Automatic Speech Recognition Systems in an under-resourced minority language (Besacier et al., 2014). In this section, we highlight challenges that are specific to the Ngiemboon language.

4.1 Lack of adequate data

Deep Learning algorithms used for training ASR acoustic models in high-resourced languages have been yielding very encouraging results with the decrease of the Word Error Rate (WER), the common metric of the performance of a Speech Recognition Systems. “DNN frameworks, however, typically require a very large amount of data, making them less useful for the low-resource scenario typically encountered with endangered languages” (Imerson et al., 2018). Gauthier (2018) seems to agree with this statement. Citing another resource, she highlights three main resources needed for the development of a state-of-the-art Automatic Speech Recognition system:

- A large text corpus (10 to 100 k words)

A large audio corpus (10 to 100 hours)

- A substantial lexical dictionary with a phonetic transcription of words.

This is useful for both language and acoustic modeling.

To the best of our knowledge, none of these resources exist or are available for the Ngiemboon language. The lack of a significant quantity of speech as well as text and audio data limits access to new machine learning algorithms that enable the development of state-of-the-art Speech Recognition Systems. It is important to note that existing corpora and data collection are integral to ASR development in any language (Besacier et al., 2014).

The development of the corpus by itself may present serious challenges in the case of the Ngiemboon language. Although the amount of data available on the web for many languages (high-resource languages as well as some under-resourced languages) is on the rise, under-resources languages like the Ngiemboon language with no existing corpora and data collection do not experience this growth. Our query on the web returned only a text of about 22k of size, collected as part of the Crubadan project carried out by Scannell (2007). Data collection in this case should anticipate higher costs, a significant amount of time, and the

availability of innovative techniques and approaches, assuming there is manpower available.

Notwithstanding significant efforts and progress made in speech data collection even for high-resource languages, "we have barely scratched the surface in sampling the many kinds of speech, environments, and channels that people routinely experience. We currently provide our automatic systems only a very small fraction of the number of materials that humans utilize to acquire language. If we want our systems to be more powerful and to understand the nature of speech itself, we need to make more use of it and label more of it. Well-labeled speech corpora have been the cornerstone on which today's systems have been developed and evolved. However, most of the large quantities of data are not labeled or poorly labeled, and labeling them accurately is costly" (Huang et al., 2014). This further highlights challenges related to the development of large speech corpora, and the quality of the corpus essential to the development of better ASR systems.

4.2 Language typology

The Ngiemboon language, like many other Bamileke languages of Cameroon, has very complex linguistic characteristics. "One of the most complex aspects of Eastern Grassfields languages is the quantity of tonal perturbations … Even more complex are the many tonal morphemes that affect verb roots in the complicated verbal constructions. While Eastern Grassfields languages are noteworthy for their lack of productive verbal suffixes with segmental material, they more than make up for this lack by the number of tonal morphemes that surround the verb. The presence of these many tonal morphemes is only revealed by the vast variety of surface tones found on verb roots in their various verbal constructions" (Anderson, 2001). Serious computational challenges are likely to emerge given the complexity of the tonal and grammatical features inherent in the Ngiemboon language. Although there has been successful computation of supra-segmental features (tone for instance) in some Asian languages (Chen et al., 2018), it is still hard to tell if neural network algorithms that yielded these positive results would produce the same level of satisfaction with the Ngiemboon language, because "the tonal system of the Grassfield Bantu Languages, in particular, is known to be among the most complicated in the world" (Anderson, 1991). Furthermore, this tonal complexity may not have told all the stories that it has to tell, despite significant descriptive linguistic studies that have been carried out so far, and it "will likely still be some time until the exact nature of Eastern Grassfields tonal perturbations is fully understood" (Anderson, 2008). This is to suggest that further and in-depth research is needed in this area to help provide answers to questions. This will only make the automation process of the language challenging.

The tonal system is not the only complex linguistic feature of this language. It has a noun class system, and many significant morphological changes stemming from the syllabic nasal prefixes and/or floating tones in the language. The authors of this paper do not know of any language with such morpho-phonological characteristics where natural language processing or speech recognition research has been carried out. They anticipate that these inherent linguistics characteristics in the language might present computational challenges.

4.3 Sociolinguistic challenges

An Ethnologue report suggests that most Ngiemboonphones speak other languages such as French, Pidgin English, or Ngiemboon neighboring languages. Consequently, many of these speakers are multilingual and frequently use code-switching in their regular conversations. The Online Merriam Webster dictionary defines code-switching as "the switching from the linguistic system of one language or dialect to that of another". This is generally observed between two bilingual speakers that share the same linguistic code. "In voice communication, many East Africans rapidly code-switch (switch between languages). This is usually done multiple times per sentence, throughout an interaction, and usually between English and another language" (Cvitkovic, 2018). This daily interaction of more than one language poses some challenges to the process of building an adequate Speech Recognition System. In fact, "The development of Automatic Speech Recognition (ASR) for code-switched speech is a current research challenge and is constrained by the difficulty of obtaining representative data for acoustic and language model training" (Ewaldvan and Niesler, 2016). This would be true for the Ngiemboon language as well.

4.4 Economic challenges

Finally, Ngiemboon speakers live mostly in rural areas where they make their living by traditional agriculture. Many also live in various towns and cities where their main activity is small trade, and they are involved in several small-scale commercial activities. In other words, the Ngiemboon language lacks an industrialization status, which presupposes that it may not have a major economic impact/advantage at the moment. This could be a discouraging factor in the development of an ASR system for this language.

5. Developing an Automatic Speech Recognition system in Ngiemboon: Perspectives

Despite the challenges discussed in the previous section, advances in machine learning and other technologies have laid down the path needed to build state-of-the-art Automatic Speech Recognition Systems in under-resourced languages, with a reasonable word error rate. The Ngiemboon language could benefit from some of these advancements. An example of such advances is Deep Learning technologies. Below the authors highlight some insightful technological pathways that will be explored as they develop speech technologies in Ngiemboon. We will not however be discussing in this paper Deep Learning architectures that have been successfully applied to ASR in under-resourced languages. Some of these are mentioned here only as a point of reference. A detailed survey of these is explored at length by Sailor et al., (2018)

5.1 LIG-Aikuma application and data collection

Automatic Speech Recognition systems are built around three pillars: acoustic models, language models, and a pronunciation lexicon. There is a strong correlation between the performance of these, and the amount of training data used. The performance quality if better with more data.

For a very long time in the past, most researchers (linguists, computational linguists, phoneticians …) used microphones and/or recorders for collecting speech data. Today there are new opportunities offering scalable networked devices that make the data collection task less tedious and less costly. A great example is the LIG-Aikuma application, an extension of Aikuma, "a mobile app that is designed to put the key language documentation tasks of recording, respeaking, and translating in the hands of a speech community… It collects recordings, respeakings, and interpretations, and organizes them for later synchronization with the cloud and archival storage… Recordings are stored alongside a wealth of metadata, including language, GPS coordinates, speaker, and offsets on time-aligned translations and comments" (Bird et al., 2014). This application is open source, freely downloadable on any android based smartphone or mobile device. In fact, "the application LIG-AIKUMA has been successfully tested on different devices (including Samsung Galaxy SIII, Google Nexus 6, HTC Desire 820 smartphones and a Galaxy Tab 4 tablet), and can be downloaded from a dedicated website … Originally intended for language documentation and data collection in the field, the app has also been useful for collecting speech for technological development purposes targeting under-resourced languages" (Besacier et al., 2019). In addition to its availability for free download, the LIG-Aikuma application is a great innovation in the area of speech data collection and offers different speech collection modes. It is a great tool that has the potential to support the collection of large quantity of data and may also help with long-term archival of the data collected. Large quantity of data is an absolute prerequisite for Deep Learning Architectures. The LIG-Aikuma application is therefore a great data collection tool available for Deep Learning based ASR pioneering research in Ngiemboon.

5.2 Data augmentation

A team of Google Brain Researchers recently made the following observation: "Deep Learning has been applied successfully to Automatic Speech Recognition (ASR), where the focus of research has been designing better network architectures, for example, DNN (Deep Neural Networks), CNNs (Convolutional Neural Networks), RNNs (Recurrent Neural Network) and end-to-end models. However, these models tend to overfit easily and require large amounts of training data. Fortunately, Data augmentation has been proposed as a method to generate additional training data for ASR" (Park et al. 2019). In other words, these new machine learning algorithms (neural networks) offer new perspectives for the development of state-of-the-art Speech recognition systems and have been applied to under-resourced languages with great success. An example includes the LSTM (Long Short-term Memory) models that exhibit better refinements over standard recurrent neural networks (Gauthier, 2018). The development of language technologies in the Ngiemboon language could certainly benefit from these. Even where and when there might be a relative shortage of data, Data Augmentation may prove to be a useful technology in the process. "Data augmentation is a common strategy adopted to increase the quantity of training data" (Povey et al., 2015). Some data augmentation techniques include "augmenting artificial data for low resource speech recognition tasks, adapting vocal tract length normalization, synthesizing noisy audio via superimposing clean sound with noisy audio signals, applying speed perturbation on raw audio for LVSCR tasks, making use of an acoustic room simulator, and studying data augmentation for keyword spotting" (Dossman, 2019). Additionally, the SpecAugment technique developed by Google Brain will also help in the data collection process (Park, et al. 2019).

6. Conclusion

The Ngiemboon language is a minority, under-resourced language spoken in the Western region of Cameroon (Africa). Although it does not enjoy a strong economic, scientific, or political status, there are compelling reasons to carry out Deep Learning-based ASR research and development in this language. As exciting as it might be, this endeavor would undoubtedly be challenging and will face several hurdles, many of which are highlighted in this paper. However, recent developments in Artificial Intelligence and other new technologies are offering new opportunities and perspectives that were not readily available decades ago. It is feasible and exciting to engage in the development of a state-of-the-art Automatic Speech Recognition system in this language. The benefits are significant, and the stakes are high. The authors of this paper encourage continuous research along these lines. They hope to complete this journey towards developing a speech corpus, and full-blown ASR system in this language and watch for its outcomes.

References

Ngefac, Aloys. 2010. Linguistic Choices in Postcolonial Multilingual Cameroon; Nordic Journal of African Studies 19 (3): 149–164.

Anderson, Stephen C. 1976a. A Phonology of Ngyemboon-Bamileke. Yaoundé, Cameroon: SIL.

Anderson, Stephen. (ed.) 1991. Tone in five languages of Cameroon. (SIL Publication in Linguistics 102). Dallas: Summer Institute of Linguistics and the University of Texas at Arlington.

Anderson, Stephen. 2001. Phonological Characteristics of Eastern Grassfields Languages. In Nguessimo M. Mutaka and Sammy B. Chumbow, ed. Research Mate in African Linguistics: Focus on Cameroon, 33-54.

Anderson, Stephen. 2008. A phonological Sketch of Ngiemboon _Bamileke. SIL-Cameroon.

Berment, Vincent. 2004. Méthodes pour informatiser des langues et des groupes de langues peu dotées. Ph.D. Thesis, J. Fourier University – Grenoble I.

Besacier, Laurent; Barnard, Etienne; Karpov, Alexey ; Schultz, Tanja. 2014. Automatic Speech recognition for under-resourced languages: a survey. Speech Communication. Volume 56; 85-100, UK.

Besacier, L., Gautier, E., Voisin, S. 2019. Lessons learned after the development and use of a data collection app for language documentation (ligaikuma). International Congress of Phonetic Sciences ICPhS, Melbourne, Australia.

Bird, Steven; Hanke, Florian; Adams, Oliver; Lee, Haejoong. 2014. Aikuma: A Mobile App for Collaborative Language Documentation. Proceedings of the 2014 Workshop on the Use of Computational Methods in the Study of Endangered Languages, pages 1–5.

Chen Charles ; Bunescu, Razvan; Xu, Li ; Liu, Chang. 2018. Tone Classification in Mandarin Chinese Using Convolutional Neural Networks. 10.21437 Interspeech. 2016-528; Wiley-IEEE Press, Hoboken, NJ.

Cvitkovic, Milan. 2018. Some Requests for Machine Learning Research from the East African Tech Scene. Proceedings of NIPS; Workshop on Machine Learning for the Developing World.

Dossman, Christopher. 2019. Google Brain Unveils a Simple Data Augmentation Method for Speech Recognition. https://medium.com.

Echu, George. 2004. The Language Question in Cameroon. Linguistik online. V.18 28(1):114-133.

Ewaldvan der Westhuizen; Thomas Niesler. 2016. Automatic Speech Recognition of English-isiZulu Code-switched Speech from South African Soap Operas; Procedia Computer Science 81; 121 – 127.

Gauthier, Elodie. 2018. Collecter, Transcrire, Analyser : quand la machine assiste le linguiste dans son travail de terrain. Ph.D Dissertation, Grenoble, France.

Huang, Xuedong; Baker, James; Reddy, Raj. 2014. A Historical Perspective of Speech Recognition. Communications of the ACM. 57. 94-103. 10.1145/2500887.

Imerson, Robbie; Simha, Kruthika; Ptucha, Raymond ; Prud'hommeaux, Emily. 2018. Improving ASR Output for Endangered Language Documentation. 182-186. 10.21437/SLTU.2018-38.

Juan, B. and Rabiner, Lawrence. 2005. Automatic Speech Recognition - A Brief History of the Technology Development.

Li, Deng; O'Shaughnessy, Douglas. 2003. Speech Processing: A Dynamic and Optimization-Oriented Approach. CRC Press (2003)

Moraa, Hilda. 2012. How Mobile technology has been used to create an impact in Cameroon. iHUB Internet blog (https://ihub.co.ke/blogs).

Park, Daniel et al. 2019: SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. arXiv:1904.08779v2 [eess.AS].

Povey, Daniel et al. 2015. Audio Augmentation for Speech Recognition. Interspeech.

Sailor, Hardik; Patil, Ankur; Patil, Hemant. 2018. Advances in Low Resource ASR: A Deep Learning Perspective. 162-166. 10.21437/SLTU.

Scannell, Kevin. 2007. The Crúbadán Project: Corpus building for under-resourced languages. Building and Exploring Web Corpora. Proceedings of the 3rd Web as Corpus Workshop.

Sneider, Keith L. 2017. Tone Analysis for Field Linguists. SIL International.

Tadadjeu, Maurice. 2004. Language, Literacy, and Education in African Development: A Perspective from Cameroon. SIL-Cameroon.

Vajpai, Jayashri; Bora, Avnish. 2016. Industrial Applications of Automatic Speech Recognition Systems. Int. Journal of Engineering Research and Applications Vol. 6, Issue 3, pp.88-

NITK-IT_NLP@NSURL2019: Transfer Learning based POS Tagger for Under Resourced Bhojpuri and Magahi Language

Anand Kumar M
Department of Information Technology
National Institute of Technology Karnataka (NITK), Surathkal
Mangalore, India
m_anandkumar@nitk.edu.in

Abstract

Part-of-Speech (POS) tagging is the primary step in the language processing task and also known to perform well automatically with a massive amount of training data. But the POS annotated training data are not available for most of the languages. The languages which do not have sufficient resources to build statistical Natural Language Processing (NLP) applications are called low-resource languages. This paper presents the machine learning-based POS tagging for low resource languages Bhojpuri and Maghali. The work is submitted to the Shared task on Low-level NLP Tools for Bhojpuri Language and Magahi Language at NSURL 2019. We develop a traditional feature-based SVM method and transfer learning-based sequence tagger using new BERT embedding, which enables better generalization to unseen words and provides regularization. The results with given minimal amounts of POS annotated data on Bhojpuri and Maghali languages show that our proposed architecture outperformed the results of the other participants and achieved the new state-of-the-art POS tagger.

1 Introduction

Part-of-Speech tagging is one of the essential stages in language processing applications. POS tagger and tagged corpus are necessary for natural language processing (NLP) to support advanced researches such as parsing, language translation, and speech recognition. If languages consist of considerable resources in terms of data, then the less engineering of hand-crafted rules is enough for robust and better performance. At the same time, the existing NLP tools are trained over large annotated corpora using machine learning techniques. But these resources are not available for most of the languages. Usually, the languages that have received relatively less attention from NLP are less popular due to their lack of available resources and are often called low-resource languages. In this work, we present methods for automatically building a POS tagger for low-resource language Bhojpuri and Maghali with minimal need for human annotation. It is difficult for researchers to produce significant resources for low-resource languages without continuous funding.

Bhojpuri is a less-resourced Indo-Aryan language of the Asian continent spoken by the western Bihar and eastern Uttar Pradesh of India and the Terai region of Nepal. Bhojpuri is sociolinguistically considered one of the Hindi dialects[1]. Magahi language is also known as Magadhi, is a language spoken in Bihar, West Bengal and Jharkhand states of India. It is also an under-resourced language and has a vibrant and old tradition of folk songs and stories[2].

There are presently between six and seven thousand languages spoken in the world (Lewis, 2009; Nettle, 1998; Wagner et al., 1999), but research in Natural Language Processing (NLP) focuses on only a small number of language. The number of internet users in a country is proportional to the regional language usage and resources available. The development of NLP applications of low-resource languages helps to increase the Internet usage of the particular region.

Research into language-independent NLP methods is desperately needed because they are appropriate in low-resource settings, and such techniques easily applied to many low-resource languages at once. The under-resourced languages can use unsupervised learning, transfer learning, and joint multilingual or polyglot learning for building NLP applications. Unsupervised feature

[1] https://en.wikipedia.org/wiki/Bhojpuri_language
[2] https://en.wikipedia.org/wiki/Magahi_language

extraction and clustering approaches used in the first learning model to build Statistical NLP applications for less-resourced languages. The variations of transfer learning include cross-lingual transfer learning, zero-shot learning, and one-shot learning (Tsvetkov). Cross-lingual transfer learning converts the resources and models from the resource-rich source language to under-resourced target language. Zero-shot learning trains a model in one domain and conceives that it generalizes in the other domain of under-resourced languages. One-shot learning trains a model in one domain and uses only a few examples from an under-resourced domain to adapt it. Transfer learning, unfortunately, only works well for closely related languages. Joint learning of resource-rich and resource-poor languages tried to provide universal representation for languages.

2 Related Works

For resource-poor languages, Feldman, Hana, and Brew (Feldman et al., 2006; Hana et al., 2004) described a method for creating taggers by combining a POS tagger and morphological analyzer. The POS tagger and morphological analyzer for closely-related source languages are helped to produce the tools for a low-resource target language. The drawback of this approach is that it is unfortunately applicable for closely related languages. Das and Petrov (Das and Petrov, 2011) proposed a new cross-lingual tagging using projected tags, and these tags are regularized using graph-based label propagation. Cross-lingual projection annotation model uses parallel corpora to bootstrap a POS tagging process without significant annotation efforts for a less-resourced language. Word-alignment (Nichols and Hwa, 2005; Yarowsky et al., 2001), and word-embedding (Adams et al., 2017) models used in bilingual and multilingual-based tagging where at least there is one resource-rich language which can help in numerous borrowings. Garrette et al. (Garrette and Baldridge, 2013; Garrette et al., 2013) explored building automatic POS taggers from tag dictionaries which created using human annotators. Unsupervised models have received perhaps the most attention for POS tagging (Johnson, 2007). The main difficulty with this unsupervised model is evaluation, where the induced word clusters and gold POS tag classes (Christodoulopoulos et al., 2010) need to compare quantitatively. SVMs widely applied for Indic language processing tasks like POS tagging, Chunking, and Morphological processing (Dhanalakshmi et al., 2009; Velliangiri et al., 2010).

BERT stands for Bidirectional Encoder Representations from Transformers (Devlin et al., 2018), which is devised to pre-train deep bidirectional representations from an unlabeled corpus by combining both left and right context in all layer. It has achieved significant progress in transfer learning for natural language understanding using the transformer architecture. The Bhojpuri POS tagged data (Singh and Jha, 2015) has been developed by using BIS guidelines. POS tagger, monolingual corpus, and Morphological Analyser are also available for Magahi language (Kumar et al., 2016). The Magahi corpora were created from blogs and stories and annotated using BIS tagset (Kumar et al., 2014).

3 POS tagging for Bhojpuri and Magahi

In the NSURL shared task, we have developed two different methods for POS tagging the Bhojpuri and Magahi languages. This section explains the data set description and the detailed methodology developed for the shared task.

3.1 Data set description

Table.1 shows the statistics of the Bhojpuri and Magahi POS data set given by the task organizers. Both language sentences were POS tagged using the Bureau of Indian Standards (BIS) annotation scheme, which is a common standard of annotation for Indian languages. Compared with the Bhojpuri tagset, Magahi consists of more tags. Bhojpuri words tagged with Fine-grained tags and Magahi words annotated with course-grained tags. The Bhojpuri language contains a more average number of words per sentence compared with the Magahi language.

3.2 Methodology

The organizers give sequence labeled POS training data in the word per line fashion. Test data provided in the same format without the POS labels. We have used two different methods to develop the POS tagger for Bhojpuri and Magahi. Figure 1 shows the methodology of the proposed model. The first method based on the common features with the Support Vector Machine (SVM) classi-

Table 1: Data set Description

Data set Description	Bhojpuri		Magahi	
	Train	Test	Train	Test
Number of sentences	4500	532	4575	604
Number of tokens	94686	10582	61431	8204
Avg Sentence length	21.04	19.89	13.43	13.58
POS-Tag set size	33		18	

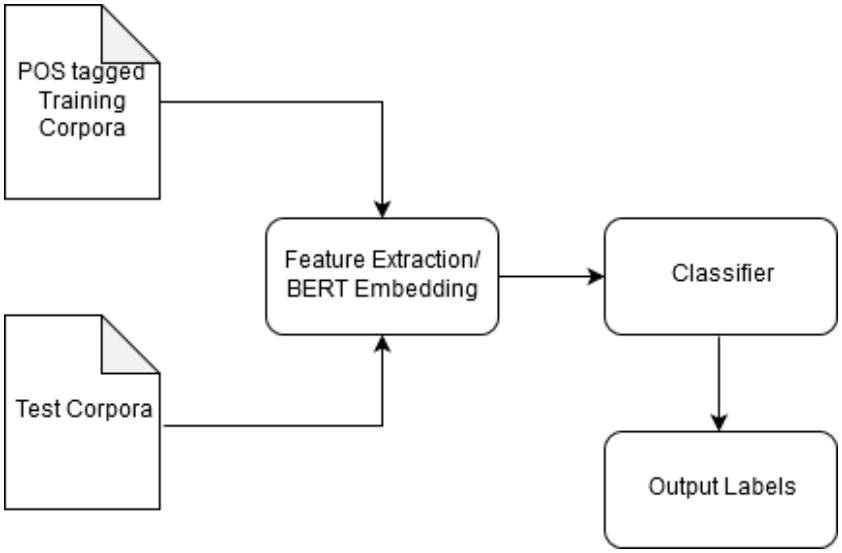


Figure 1: Methodology.

fier. This model experiment with combinations of the character bigrams, trigrams, 4-grams, 5-grams, and the full word as features. We have also considered the previous two words and the next two words as additional features. These proposed features can extract the salient features from the text. We have used the SVMLight tool (Giménez and Márquez, 2004; Joachims, 1999) for classifying and tuned the C parameter values based on cross-validation.

In the transfer-learning based method, we have used the BERT multilingual pre-trained embedding in the initial layer. The BERT embeddings consider each token and the sentence from the data set and assign the contextual representation for each token. The logits layer used in the last neuron layer of neural network for the classification task. The parameters settings for BERT given as follows, 12-layer, 768-hidden, 12-heads, 110M parameters, batch_size=8, Adam Optimizer and, Learning rate = 0.0001 with final Cross-Entropy Loss.

3.3 Experiments and Results

The parameters of the learning models are fixed using standard validation techniques. For tuning the SVM parameters, we have used 10-fold cross-validation. In the case of transfer learning randomly selected 10 percent of the training data are considered as validation data and the accuracies are reported in Table 2. Table 3 and 4 show the accuracies of the developed POS tagger achieved on the shared task. We have submitted our runs in the team name of "NITK-IT_NLP" and "SUB1" represents the conventional feature-based SVM classifier and "SUB2" refers to the transfer learning model.

From the accuracy tables, it is clear that the SVM based method worked perfectly for Bhojpuri, and the transfer learning model worked well for the Magahi language. Interestingly, the accuracies are indirectly proportional to the tagset size of the language (Usually, the accuracy is comparably less for the language with fine-grained tagset i.e. Bhojpuri language). If we compare the accuracies of both languages, the method which gives good accuracy for one language is provides less accu-

Table 2: Validation Accuracy

Methods	Bhojpuri	Magahi
WordFeat+CharFeat+SVM (10 Fold)	94.38	80.11
TransferLearning-BERT(Random10Per)	90.1	89.0

racy in another language. It is right in vice-versa also. The reason is the number of tags in the tagset and an average number of words in a sentence.

Table 3: Accuracy of Magahi Language

Rank	Team / Run	F1 Score
1	NITK-IT_NLP_SUB2	0.79
2	the_illiterati_mag_1	0.77
2	the_illiterati_mag_2	0.77
3	the_illiterati_mag_3	0.74
4	NITK-IT_NLP_SUB1	0.73

4 Conclusion and Future Scope

Most of the research in Computational Linguistics and NLP focuses on languages that have a massive amount of text corpora. State-of-the-art NLP models also require large amounts of training data from which it can learn parameters and better co-efficient for the machine learning model. Under-resourced languages or less-resourced languages are languages which are lacking large digital text and insufficient handcrafted linguistic resources for building statistical NLP applications. Here we have presented the two POS tagging approaches developed and submitted for the Shared task on Low-level NLP Tools for Bhojpuri Language and Magahi Language at NSURL 2019. The sequence labeling formulation methods acted as a benchmark for fully supervised POS tagging. The proposed SVM based and transfer learning-based models outperform the other submissions by the participants and achieved the new state-of-the-art. It proves the need for transfer learning to the under-resourced languages. Detailed error analysis and tag specific accuracy are the other directions of future research. The research efforts exist that explore which type of linguistic features in the language and other rich-resourced languages contribute to accurate part-of-speech tagging for the low resourced languages under investigation.

Table 4: Accuracy of Bhojpuri Language

Rank	Team / Run	F1 Score
1	NITK-IT_NLP_SUB1	0.95
1	the_illiterati_bho_3	0.95
2	the_illiterati_bho_1	0.93
3	the_illiterati_bho_2	0.92
4	NITK-IT_NLP_SUB2	0.89

Acknowledgment

We want to thank the shared task organizers for organizing the workshop for under-resourced languages.

References

O. Adams, A. Makarucha, G. Neubig, S. Bird, and T. Cohn. 2017. Cross-lingual word embeddings for low- resource language modeling. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1*, pages 937–947.

C. Christodoulopoulos, S. Goldwater, and M. Steedman. 2010. Two decades of unsupervised pos induction: How far have we come? In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 575–584. Association for Computational Linguistics.

D. Das and S. Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. *in ACL, pp*, pages 600–609.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. preprint, arXiv.

V. Dhanalakshmi, P. Padmavathy, Anand Kumar, Soman M., Rajendran K. P., and S. Chunker for Tamil. 2009. Chunker for tamil. In *ARTCom 2009 - International Conference on Advances in Recent Technologies in Communication and Computing*.

A. Feldman, J. Hana, and C. Brew. 2006. A cross-language approach to rapid creation of new morpho-syntactically annotated resources. *in Proceedings of LREC, pp*, pages 549–554.

D. Garrette and J. Baldridge. 2013. Learning a part-of-speech tagger from two hours of annotation. *in Proceedings of NAACL-HLT, pp*, pages 138–147.

D. Garrette, J. Mielens, and J. Baldridge. 2013. Real-world semi-supervised learning of pos-taggers for low-resource languages. *ACL*, 1:583–592.

Jesús Giménez and Lluís Márquez. 2004. Svmtool: A general pos tagger generator based on support vector machines. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*.

J. Hana, A. Feldman, and C. Brew. 2004. A resource-light approach to russian morphology: Tagging russian using czech resources. *in EMNLP, pp*, pages 222–229.

T. Joachims. 1999. Making large-scale svm learning practical. In B. Schlkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT-Press.

M. Johnson. 2007. Why doesn't em find good hmm pos-taggers? *in EMNLP-CoNLL, pp*, pages 296–305.

R. Kumar, B. Lahiri, and D. Alok. 2014. Developing lrs for non-scheduled Indian languages. In J. Mariani, editor, *Vetulani Z.* Human Language Technology Challenges for Computer Science and Linguistics. LTC 2011. Lecture Notes in Computer Science, vol 8387. Springer, Cham.

R. Kumar, Atul Kr. Ojha, B. Lahiri, and D. Alok. 2016. Developing resources and tools for some lesser-known languages of india. *Regional ICON(regICON)*, 2016.

M. P. Lewis. 2009. *Ethnologue: Languages of the world sixteenth edition*. ethnologue. com, Dallas, Tex SIL International. Online version.

D. Nettle. 1998. Explaining global patterns of language diversity. *Journal of anthropological archaeology*, 17(4):354–374.

C. Nichols and R. Hwa. 2005. Word alignment and cross-lingual resource acquisition. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 69–72. ACL.

S. Singh and G. N. (2015 Jha. 2015. Statistical tagger for bhojpuri (employing support vector machine). *In Advances in Computing, Communications and Informatics(ICACCI)International Conference*, pages 1524–1529.

Yulia Tsvetkov. Opportunities and challenges in working with low-resource languages. *Slides Part-1*.

D. Velliangiri, M. Anand Kumar, R. U. Rekha, K. P. Soman, and S. Rajendran. 2010. Grammar teaching tools for tamil language. *2010 International Conference on Technology for Education*, 4.

D. A. Wagner, R. L. Venezky, and B. V. Street. 1999. *Literacy: An international handbook*. Westview Press Boulder.

D. Yarowsky, G. Ngai, and R. Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *01, Association for Computational Linguistics, Stroudsburg, PA, USA, p*, pages 1–8, HLT. Proceedings of the First International Conference on Human Language Technology Research.

The_Illiterati: Part-of-Speech Tagging for Magahi and Bhojpuri without Even Knowing the Alphabet

Thomas Proisl
Institute of Cognitive Science
Osnabrück University
thomas.proisl@uos.de

Peter Uhrig
English and American Studies
FAU Erlangen-Nürnberg
peter.uhrig@fau.de

Philipp Heinrich
Comp. Corpus Linguistics
FAU Erlangen-Nürnberg
philipp.heinrich@fau.de

Andreas Blombach
Comp. Corpus Linguistics
FAU Erlangen-Nürnberg
andreas.blombach@fau.de

Sefora Mammarella
Romance Studies
FAU Erlangen-Nürnberg
sefora.mammarella@icloud.com

Natalie Dykes
Computational Corpus Linguistics
FAU Erlangen-Nürnberg
natalie.mary.dykes@fau.de

Besim Kabashi
Computational Corpus Linguistics
FAU Erlangen-Nürnberg
besim.kabashi@fau.de

Abstract

In this paper, we describe the part-of-speech-tagging experiments for Magahi and Bhojpuri that we conducted for our participation in the NSURL 2019 shared tasks 9 and 10 (Low-level NLP Tools for (Magahi|Bhojpuri) Language). We experiment with three different part-of-speech taggers and evaluate the impact of additional resources such as Brown clusters, word embeddings and transfer learning from additional tagged corpora in related languages. In a 10-fold cross-validation on the training data, our best-performing models achieve accuracies of 90.70% for Magahi and 94.08% for Bhojpuri. Accuracy increased to 94.79% for Magahi and dropped to 78.68% for Bhojpuri on the test data.

1 Introduction and Related Work

Magahi and Bhojpuri are two of the three principal languages of the Bihari group (Maithili being the third). There are competing categorizations of the Bihari group within the Indo-Aryan languages (see Grierson, 1903; Cardona, 1974; Jeffers, 1976). While there are few Magahi speakers outside of Southern Bihar, Bhojpuri is spoken in parts of two Indian states, Western Bihar and Eastern Uttar Pradesh, and the Southwest of Nepal. According to the 2011 census, about 51 million people in India stated Bhojpuri as their mother tongue, and about 13 million did so for Magahi. However, these numbers may seriously underestimate the actual number of speakers, since speakers of both languages often name Hindi as their first language – the language used in schools, courts, and other public institutions (Verma, 2003b, p. 547).

Despite these numbers, comparatively few linguistic resources and NLP tools currently exist for both languages, with most of the scarce attention having gone towards Bhojpuri (e.g. Ojha et al., 2015).

It is beyond the scope of this paper and our own expertise to describe both languages in detail (but see, e.g., Verma, 2003b,a). Among the features which appear pertinent to part-of-speech tagging of Magahi and Bhojpuri are SOV order, rich verb morphology, the extensive use of postpositions, and the unusual agreement system of Magahi (where the verb has to agree with subject and object simultaneously).

Table 1 gives an overview of the two datasets of the shared task. While the training set for Bhojpuri is much larger, it also features a more fine-grained tagset.

	Magahi	Bhojpuri
training	61.435	94.692
test	8.205	10.582
tagset size	18	33

Table 1: Sizes of the training and test sets and of the tagsets.

2 Strategies and Systems

2.1 Part-of-Speech Taggers

We experiment with three different, freely available part-of-speech taggers:

- SoMeWeTa (Proisl, 2018), a tagger based on the averaged structured perceptron that supports domain adaptation and can incorporate external information sources such as Brown clusters.[1]

- A BiLSTM+CRF sequence tagger by Guillaume Genthial that uses character and word embeddings and supports transfer learning.[2]

- The Stanford Tagger (Toutanova et al., 2003), which is based on a maximum entropy cyclic dependency network.[3]

2.2 Additional Resources

In addition to the training data provided by the task organizers, we use the following freely available resources:

- The Hindi UD treebank, which is based on the Hindi Dependency Treebank (HDTB; ca. 352,000 tokens; Bhat et al., 2017; Palmer et al., 2009).[4]

- A POS-tagged Magahi corpus (KMI-Mag; ca. 46,000 tokens) and a corpus of untagged Magahi texts (ca. 2.8 million tokens).[5]

- Wikimedia dumps for Hindi (ca. 34.7 million tokens) and Bihari (ca. 700,000 tokens).[6] We extract the plain text using wikiextractor[7] and tokenize and sentence-split it using the ICU tokenizer via polyglot.[8]

- Brown clusters (Brown et al., 1992) computed from the tokenized Wikimedia dumps and the untagged Magahi corpus (1000 clusters, minimum frequency 5).[9]

- Pre-trained fastText embeddings for Hindi and Bihari[10]

The additional tagged Magahi corpus (KMI-Mag) is annotated with a tagset consisting of 35 tags which is almost identical to the 33-tag tagset used in the Bhojpuri corpus. KMI-Mag uses three tags that do not occur in the Bhojpuri data (`V_VM_VF`, `V_VM_VNF` and `V_VM_VNP`) and misses one tag that is used for Bhojpuri (`RD_ECH_B`). For our transfer learning experiments targeting Bhojpuri, we simply convert the three verb tags to `V_VM`. For targeting Magahi, we map the 35 tags to UD tags.

2.3 Experiments using SoMeWeTa

The distinctive features of SoMeWeTa are its ability to leverage additional resources and its transfer learning or domain adaptation capabilities. Consequently, we focus on these two aspects in our experiments.

For Bhojpuri, we experiment primarily with the Brown clusters computed from the Hindi and Bihari Wikimedia dumps and the untagged additional Magahi corpus (cf. section 2.2). Our cross-validation experiments show that the Brown clusters have a small positive effect with the best results being obtained by Brown clusters computed from the union of all three additional corpora (cf. Table 2). With KMI-Mag we have a corpus of a closely related language that is annotated with an almost identical tagset (cf. section 2.2). However, pretraining on that and then adapting to Bhojpuri seems to have no noticeable effect.

For Magahi, we experiment with a wide range of transfer learning settings in addition to the different Brown clusters:

- Pretraining on one of KMI-Mag, HDTB or the Bhojpuri dataset (mapped to UD tags).

- Pretraining on all possible combinations of KMI-Mag, HDTB and the Bhojpuri dataset (using the concatenation of these resources).

- Longer pretraining chains where we start with HDTB and adapt to one or two other resources before we make the final adaptation to Magahi.

The best results are obtained by using Brown clusters computed from the Hindi Wikimedia dumps

[1] https://github.com/tsproisl/SoMeWeTa

[2] We use the slightly modified version by Riedl and Padó (2018): https://github.com/riedlma/sequence_tagging

[3] https://nlp.stanford.edu/software/tagger.html

[4] https://github.com/UniversalDependencies/UD_Hindi-HDTB/tree/master

[5] https://github.com/kmi-linguistics/magahi

[6] https://dumps.wikimedia.org

[7] http://medialab.di.unipi.it/wiki/Wikipedia_Extractor

[8] http://polyglot-nlp.com/

[9] We use the implementation by Liang (2005): https://github.com/percyliang/brown-cluster

[10] https://fasttext.cc/docs/en/crawl-vectors.html

model	accuracy
No additional resources	91.62 (±0.97)
Hindi Brown clusters	91.79 (±1.00)
Bihari Brown clusters	91.60 (±1.01)
Magahi Brown clusters	91.69 (±0.93)
Hindi+Magahi Brown clusters (hi+mag)	91.99 (±0.83)
Hindi+Bihari+Magahi Brown clusters (hi+bh+mag)	*92.04 (±0.80)*
KMI-Mag → Bhojpuri, hi+mag	92.03 (±0.90)
KMI-Mag → Bhojpuri, hi+bh+mag	92.06 (±0.94)

Table 2: Bhojpuri results for SoMeWeTa. We report the mean accuracies and 95% confidence intervals of a 10-fold cross-validation on the training data. The model that we submitted to the shared task is set in italics.

and the untagged additional Magahi corpus. As for Bhojpuri, transfer learning does not seem to have any noticeable effect (cf. Table 3).

2.4 Experiments using the BiLSTM-CRF tagger

Neural networks with a BiLSTM-CRF architecture achieve POS-tagging results close to the current state of the art.[11] In our experiments, we focus less on the hyperparameters of the network but rather on the effects of our additional resources. We try out both the Hindi and Bihari fastText embeddings. Since the Bihari embeddings do not perform significantly better than the Hindi embeddings (cf. Table 4) and the Hindi embeddings cover a much larger vocabulary (15.3 million words instead of 8.9 million), we use the Hindi embeddings for our further experiments. In the following, we make use of the tagger's transfer learning abilities and pretrain the models on HDTB or KMI-Mag. The BiLSTM-CRF tagger seems to benefit more from the transfer learning setting than SoMeWeTa and achieves its best results for both languages with a transfer from KMI-Mag. Interestingly, the BiLSTM-CRF outperforms SoMeWeTa only on the Magahi dataset while it performs notably worse on the Bhojpuri dataset.

2.5 Experiments using the Stanford Tagger

The Stanford Log-linear Part-Of-Speech Tagger (Toutanova and Manning 2000; Toutanova et al. 2003) is a mature and stable tagger that still exhibits competitive performance. The system is feature-rich and offers a range of configuration options, the effects of which were initially not fully understood by our research group. It was thus decided to run extensive brute-force hyperparameter tuning making educated guesses about the value ranges of the various parameters. The documentation in the JavaDoc for the MaxentTagger class[12] provides the necessary information. It was decided to set the following parameters with the values or ranges given in Table 5 and Table 6.

Combining all parameters results in 76,800 parameter combinations per language. Although training and testing can be completed in approximately 2 minutes on a modern personal computer, the sheer number of parameter combinations necessitated running the experiments on High-Performance-Computing infrastructure. The setup comprised a central queue of filenames of property files that all involved clients subscribed to.

For Magahi, only two runs with all parameter combinations were performed: one with the top 80% of the training data as actual training data and the bottom 20% as test data and one with the bottom 80% as training data and the top 20% as test data. The values discussed below are the arithmetic mean of the accuracies of those two runs. As the Magahi tagset is Universal-Dependencies-compliant, it was straightforward to identify closed class words by pos tag and to supply the list to the tagger during the training phase.

For Bhojpuri, a full 10-fold cross-validation was carried out for each of the parameter combinations, so the averages discussed below are most likely more reliable than those for Magahi. Since the Bhojpuri tagset was more complicated, we decided to learn the closed class tags automatically based on the default *closedClassTagThreshold* of 40. Thus, a pos tag is only considered a closed class if it is assigned to less than 40 different words.

[11] Cf. https://aclweb.org/aclwiki/POS_Tagging_(State_of_the_art)

[12] https://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/tagger/maxent/MaxentTagger.html

model	accuracy
No additional resources	88.92 (±1.24)
Hindi Brown cluster	89.07 (±1.24)
Bihari Brown cluster	88.90 (±1.32)
Magahi Brown cluster	89.12 (±1.23)
Hindi+Magahi Brown cluster	*89.32 (±1.15)*
Hindi+Bihari+Magahi Brown cluster	89.15 (±1.17)
KMI-Mag → Magahi, Hindi+Magahi Brown cluster	89.20 (±1.10)
KMI-Mag → Magahi, Hindi+Bihari+Magahi Brown cluster	89.23 (±1.19)
Bhojpuri → Magahi, Hindi+Magahi Brown cluster	89.25 (±1.13)
Bhojpuri → Magahi, Hindi+Bihari+Magahi Brown cluster	89.18 (±1.25)
HDTB → Magahi, Hindi+Magahi Brown cluster	89.26 (±1.21)
HDTB → Magahi, Hindi+Bihari+Magahi Brown cluster	89.17 (±1.18)
HDTB+KMI-Mag → Magahi, Hindi+Magahi Brown cluster	89.22 (±1.12)
HDTB+KMI-Mag → Magahi, Hindi+Bihari+Magahi Brown cluster	89.19 (±1.23)
HDTB+Bhojpuri → Magahi, Hindi+Magahi Brown cluster	89.23 (±1.13)
HDTB+Bhojpuri → Magahi, Hindi+Bihari+Magahi Brown cluster	89.18 (±1.20)
KMI-Mag+Bhojpuri → Magahi, Hindi+Magahi Brown cluster	89.30 (±1.14)
KMI-Mag+Bhojpuri → Magahi, Hindi+Bihari+Magahi Brown cluster	89.06 (±1.19)
HDTB+KMI-Mag+Bhojpuri → Magahi, Hindi+Magahi Brown cluster	89.21 (±1.17)
HDTB+KMI-Mag+Bhojpuri, Hindi+Bihari+Magahi Brown cluster	89.20 (±1.20)
HDTB → KMI-Mag → Magahi, Hindi+Magahi Brown cluster	89.24 (±1.20)
HDTB → KMI-Mag → Magahi, Hindi+Bihari+Magahi Brown cluster	89.22 (±1.18)
HDTB → Bhojpuri → Magahi, Hindi+Magahi Brown cluster	89.27 (±1.14)
HDTB → Bhojpuri → Magahi, Hindi+Bihari+Magahi Brown cluster	89.11 (±1.17)
HDTB → Bhojpuri → KMI-Mag → Magahi, Hindi+Magahi Brown cluster	89.22 (±1.11)
HDTB → Bhojpuri → KMI-Mag → Magahi, Hindi+Bihari+Magahi Brown cluster	89.20 (±1.19)

Table 3: Magahi results for SoMeWeTa. We report the mean accuracies and 95% confidence intervals of a 10-fold cross-validation on the training data. The model that we submitted to the shared task is set in italics.

model	accuracy
Magahi (Hindi embeddings)	88,97 (±1,14)
Magahi (Bihari embeddings)	89,09 (±1,00)
HDTB → Magahi (Hindi embeddings)	89,85 (±0,99)
KMI-Mag → Magahi (Hindi embeddings)	*90,70 (±0,92)*
Bhojpuri (Hindi embeddings)	90,78 (±0,55)
Bhojpuri (Bihari embeddings)	90,80 (±0,57)
KMI-Mag → Bhojpuri (Hindi embeddings)	*91,23 (±0,68)*

Table 4: Results for the BiLSTM-CRF tagger. We report the mean accuracies and 95% confidence intervals of a 10-fold cross-validation on the training data. The models submitted to the shared task are set in italics.

Given that the training dataset is smaller than what is available for more commonly researched languages, we expected that for most thresholds, values below the default values might be more relevant than above, which is why our choice of parameter values is skewed towards smaller numbers.

For both languages, performance decreases abruptly when *rareWordThresh* is set to 1. We exclude this setting for the remainder of the analysis, since it is obviously beneficial for the tagger to treat hapax legomena as rare words. Additionally, performance was insensitive to variation in *veryCommonWordThresh* since this value is ignored by the Tagger in our case. We thus fix the threshold at 250 and use simple linear models without interaction to analyze the influence of all other variables on performance measures:

$$\text{acc.} = \beta_0 + \beta_1(\textit{unicodeshape}) + \beta_2(\textit{macro}) + \sum_{j=3}^{6} \beta_j \gamma_j + \varepsilon$$

where β_i are the coefficients, γ_j is one of the integer features (*rareWordThresh*, *curWordMinFeatureThresh*, *minFeatureThresh*, *rareWordMinFeatureThresh*), and ε is the residual error.

Accuracy for Bhojpuri reaches around $\mu \approx 93.88$ with a standard deviation of approximately 0.064 and the linear model yielding an adjusted R^2 of approximately 0.80. For Magahi, overall performance is lower ($\mu \approx 87.66$) and variation is higher ($\sigma \approx 0.51$), but this variation is well-explained by the linear model (adjusted $R^2 \approx 0.98$).

For both languages, the *macro* parameter has the most influence on accuracy. For Bhojpuri, the best *macro* is `bidirectional5words` (yielding ceteris paribus 0.09 and 0.12 better results compared to `generic` and `left3words`, respectively). For Magahi, however, `generic`

parameter	default value	value/range
closedClassTags	(none)	ADP AUX CCONJ DET NUM PART PRON SCONJ PUNCT
arch - architecture	generic	generic, left3word, bidirectional5words
arch - further unknown-words option	(none)	naacl2003unknowns
arch - unicode shapes for rare words	(none)	unicodeshapes(-2,2), unicodeshapes(-1,1), unicodeshapes(0), (none)
iterations	100	100
learnClosedClassTags	false	false
curWordMinFeatureThresh	2	1..4
minFeatureThresh	5	1..5
rareWordMinFeatureThresh	10	1..10
rareWordThresh	5	1..8
veryCommonWordThresh	250	100, 150, 200, 250

Table 5: Settings and parameters with ranges for the training of the Stanford PoS Tagger for Magahi.

parameter	default value	value/range
closedClassTags	(none)	(none)
arch - architecture	generic	generic, left3word, bidirectional5words
arch - further unknown-words option	(none)	naacl2003unknowns
arch - unicode shapes for rare words	(none)	unicodeshapes(-2,2), unicodeshapes(-1,1), unicodeshapes(0), (none)
iterations	100	100
learnClosedClassTags	false	true
closedClassTagThreshold	40	40
curWordMinFeatureThresh	2	1..4
minFeatureThresh	5	1..5
rareWordMinFeatureThresh	10	1..10
rareWordThresh	5	1..8
veryCommonWordThresh	250	100, 150, 200, 250

Table 6: Settings and parameters with ranges for the training of the Stanford PoS Tagger for Bhojpuri.

and `left3words` give better results (both approximately 1.0 accuracy points better than `bidirectional5words`). This is surprising, since according to the authors of the Stanford Tagger, "[t]he `left3words` architectures are faster, but slightly less accurate, than the `bidirectional` architectures."[13] The only viable explanation that comes to mind is that possibly the Magahi gold standard corpus was annotated with a trigram tagger without sufficient manual correction. This is in line with our observation that in the Magahi data, items that should have been classified as punctuation marks recieved dubious tags, e.g. the grave accent (`) was tagged only twice as punctuation, but was categorized as a noun five times, twice as an adposition, once as a verb and once as an auxiliary.

Examining only the respective best-performing *macro*, *rareWordThresh* explains most of the remaining variation, with a significant regression coefficient of about 0.02 for Bhojpuri and 0.07 for Magahi. However, the effect might decrease for values higher than the ones tested here (*rareWordThresh* $\in \{1, \ldots, 8\}$).

unicodeshape has a small effect on performance for Bhojpuri, where `(-1,1)` and `(-2,2)` yield an increase in performance by about 0.06 compared to `(0)` and `None`. This effect cannot be confirmed for Magahi. For both languages, performance decreases in *curWordThresh*, *curWordMinFeatureThresh*, and *rareWordMinFeatureThresh*, though the effect is negligible and not always significant. In both cases, *minFeatureThresh* does not have a significant influence on accuracy.

3 Results and Error Analysis

3.1 Bhojpuri

The overall results for Bhojpuri are delightful since they are even better than on our training data (see Table 7): Our optimized version of the Stanford tagger scored 95 points macro F_1 (94.78 accuracy), and we thus share first place with our sole competitor (team *NITK-NLP*); SoMeWeta and the BiLSTM tagger are close behind.

We omit the very large confusion matrix (33×33 and predominantly zero off the diagonal)

[13] https://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/tagger/maxent/ExtractorFrames.html

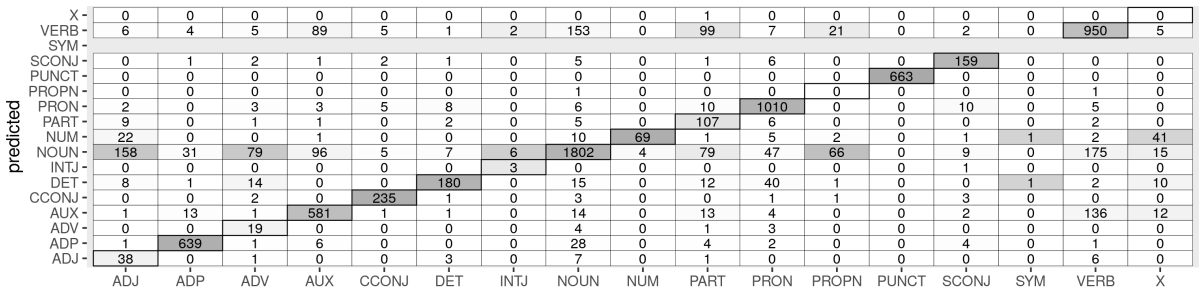


Figure 1: Confusion Matrix for SoMeWeTa predicting Magahi tags on the test data. Absolute numbers are given for all cells; shade represents recall (on the diagonal) and false positive rate, respectively. Actual labels can be found on the abscissa, predicted ones on the ordinate.

rank	submission	F_1
1	**Stanford**	95
1	NITK-NLP_SUB1	95
2	**SoMeWeTa**	93
3	**BiLSTM-CRF**	92
4	NITK-NLP_SUB2	89

Table 7: Results for Bhojpuri

and instead provide a quick summary for the Stanford tagger:[14]

- Two tags are not predicted by our tagger at all: RD_ECH_B (which appears once in the gold data and was misclassified as N_NN), and RD_UNK (classified once as N_NN and once as V_VM).
- RP_INJ appeared five times in the gold standard and was predicted correctly four times. This tag yields the worst recall (apart from the two pathological cases above).
- 30 of the 195 occurrences of RD_SYM were misclassified (recall 84.6%), mostly as N_NN (26 cases).
- Further incorrect predictions of N_NN occur for JJ (11.3% of its occurrences classified as N_NN, 85.2% recall), RB (7.7%, 89.7% recall), and N_NNP (6.4%, 92.8% recall).
- Another notable confusion is the pair V_VM (87.8% recall) and V_VAUX (86.6% recall); V_VM was predicted as V_VAUX 64 times, while V_VAUX was tagged V_VM 66 times. Finally, V_VM was predicted as N_NN 85 times.

[14]We focus on recall; precision is mostly the same as recall for all frequent labels, and higher for rare ones, since the taggers avoid predicting infrequent labels.

The results for our other submissions were very much in line with the results discussed here.[15] All in all, the errors made by our submissions are very much what one would expect: Very rare categories are sometimes misclassified, very frequent categories (such as N_NN) tend to be the go-to label for misclassifications, and similar morphosyntactic categories are confused with each other (V_VM and V_AUX, N_NN and N_NNP).

3.2 Magahi

With a macro F_1 score of only 77%, our best submissions, SoMeWeTa (78.68 accuracy) and BiLSTM-CRF (78.86 accuracy), rank second in the task of predicting Magahi tags, closely behind the submissions of one of our competing teams (see Table 8). Results are peculiar, since this is a drop of more than ten points compared to our cross-validation on the training data set and far outside our realized confidence intervals (see Table 3).

rank	submission	F_1
1	NITK-NLP_SUB2	79
2	**SoMeWeTa**	77
2	**BiLSTM-CRF**	77
3	**Stanford**	74
4	NITK-NLP_SUB1	73

Table 8: Results for Magahi

Figure 1 shows the confusion matrix for SoMeWeTa.[16] Major problems arise for tags ADJ (15.5% recall), ADV (14.8%), PART (32.5%), and PROPN and X (both 0%), since these are quite frequent categories with severe error rates. As with

[15]One notable exception is that the BiLSTM tagger did non predict the category RD_ECH at all (another hapax in the gold standard) but did include RD_ECH_B (once, incorrectly).

[16]Again, results are very similar for our other submissions.

Bhojpuri, the tagger misclassifies them as NOUNs and VERBs, which are the most frequent open classes. Moreover, the tagger frequently mistakes VERB for AUX and vice versa.

4 Conclusion

The results for Bhojpuri are very satisfying. Close to 95% accuracy on a set of 33 tags with approximately 95,000 words of training data is in line with our expectations. It is a bit disappointing, however, that mindless parameter-tuning yields the best results – but the difference may very well not be significant.

The results for Magahi are very disappointing. Since we do not know the language, it is difficult for us to pinpoint the exact reasons for the bad performance, be it an over-generalization of our taggers, a shift in the tag distribution in the test data or an issue with the annotation quality. At least, however, the use of additional resources outperforms mere parameter-tuning.

References

Riyaz Ahmad Bhat, Rajesh Bhatt, Annahita Farudi, Prescott Klassen, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, Ashwini Vaidya, Sri Ramagurumurthy Vishnu, et al. 2017. The Hindi/Urdu treebank project. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, pages 659–697. Springer.

Peter F. Brown, Vincent J. Della Pietra, Peter V. de Souza, Jennifer C. Lai, and Robert L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

George Cardona. 1974. *The Indo-Aryan languages*, 15th edition, volume 9, pages 439–450.

George Abraham Grierson. 1903. *Linguistic survey of India, Vol. V: Indo-Aryan Family, Eastern Group, Pt. II: Specimens of the Bihari and Oriya Languages*. Office of the Superintendent of Government Printing, India, Calcutta.

Robert J. Jeffers. 1976. The position of the Bihārī dialects in Indo-Aryan. *Indo-Iranian Journal*, 18(3):215–225.

Percy Liang. 2005. Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science.

Atul Ku Ojha, Pitambar Behera, Srishti Singh, and Girish N. Jha. 2015. Training & evaluation of POS taggers in Indo-Aryan languages: a case of Hindi, Odia and Bhojpuri. In *Proceedings of the 7th Language & Technology Conference (LTC 2015)*, pages 524–529.

Martha Palmer, Rajesh Bhatt, Bhuvana Narasimhan, Owen Rambow, Dipti Misra Sharma, and Fei Xia. 2009. Hindi syntax: Annotating dependency, lexical predicate-argument structure, and phrase structure. In *The 7th International Conference on Natural Language Processing*, pages 14–17.

Thomas Proisl. 2018. SoMeWeTa: A part-of-speech tagger for German social media and web texts. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 665–670, Miyazaki. European Language Resources Association.

Office of the Registrar General & Census Commissioner. 2011. 2011 census data, Data on language and mother tongue, Statement 1: Abstract of speakers' strength of languages and mother tongues.

Martin Riedl and Sebastian Padó. 2018. A named entity recognition shootout for German. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018), Volume 2: Short Papers*, pages 120–125, Melbourne.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT-NAACL 2003*, pages 252–259, Edmonton.

Kristina Toutanova and Christopher D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of EMNLP/VLC-2000*, pages 63–70.

Manindra K. Verma. 2003a. Bhojpuri. In George Cardona and Dhanesh Jain, editors, *The Indo-Aryan Languages*, pages 566–589. Routledge, London.

Sheela Verma. 2003b. Magahi. In George Cardona and Dhanesh Jain, editors, *The Indo-Aryan Languages*, pages 547–565. Routledge, London.

ST NSURL 2019 Shared Task: Semantic Question Similarity in Arabic

Mohamed Lichouri, Mourad Abbas, Besma Benaziz
Computational Linguistics Dept.,CRSTDLA
Algeria
m.lichouri@crstdla.dz
m.abbas@crstdla.dz
b.benaziz@crstdla.dz

Abed Alhakim Freihat
University of Trento
Italy
abed.freihat@unitn.it

Abstract

In this paper, we describe the solution that we propose for the shared task NSURL 2019 Semantic Question Similarity in Arabic. The proposed solution combines three approaches: lexical, statistical, and neural. The lexical approach is based on similarity measures. The statistical approach utilizes a set of binary classifiers. The neural approach uses a Siamese Deep Neural Network Model.

1 Introduction

The task in NSURL 2019 Semantic Question Similarity in Arabic shared task (Seelawi et al., 2019) is to predict the semantic similarity between questions: For a given question pair $< q_1, q_2 >$, identify if the questions q_1 and q_2 have the same meaning or not.

A question similarity system is an important component that contributes to good question answering portal. This component enables users to find answers to previously asked questions similar to their own before posting new questions.

Many Question similarity approaches have been already proposed for English (Nakov et al., 2016) and other European languages such as Spanish, French or Italian (Buscaldi et al., 2010).

For the Arabic language, there are also some question similarity proposals (Abouenour et al., 2010). were Such approaches do not give a general solution to the problem of question semantic similarity due to some limitations these systems have. For example, QARAB (Hammo et al., 2002) system does not take into consideration the understanding of the content of the question at a semantic level. AQAS (Mohammed et al., 1993) system is designed fro structured texts only. ArabiQA (Benajiba et al., 2007) and QASAL (Brini et al., 2009) Systems target factoid questions only.

		# Sentences	# Words
Train	**Ques1**	11,995	68,608
	Ques2	11,995	64,039
	QuesPairs	11,995	64,039
Test	**Ques1**	3,715	21,248
	Ques2	3,715	19,682
	QuesPairs	3,715	19,682
Total	**QuesPairs**	15,710	83,721

Table 1: Statistics of the used dataset.

The proposed system in this paper combines three approaches: lexical, statistical and neural. In the lexical approach, we use a set of text similarity measures from the text distance tools. In the statistical approach, we deploy a set of classifiers. In the neural approach, we apply a Siamese Deep Neural Network Model. We also use additional features such as punctuation and stop word filtering, normalization, stemming, and POS-tagging to enhance the final results.

The rest of the paper is organized as follows; we describe our data in Section 2 and the proposed system in Section 3. We report our experiments and results in 4 and conclude with conclusion and suggestions for future research in 5.

2 Dataset

In this work, we used the NSURL Task8(Seelawi et al., 2019) data set provided by the Mawdoo3 Team. The training data is composed of 11995 sentences. The size of the test sets is 200 questions pairs. The questions are short, ranging from 4 to 15 words each. Each sentence is annotated with the speaker dialect. In table 1, we provide some statistics on the used corpora.

3 System

The presentation of our proposed system is shown in figure 1.

In the following, we summarize the approach:

1. In parallel, run the three approaches (lexical, statistical, and neural).
2. Select the three best configurations that achieved the best performance.
3. In the third step, apply a combination of features which will give us the best model for each approach.
4. In the last step, combine two of the three models 2by2.This enables us to have a lexical-statistical combination approach and a neural approach.

3.1 Features extraction

3.1.1 ngrams features

The first features that we considered to deal with the problem of Semantic Question Similarity in Arabic, were the word and character n-grams features used in previous work such as (Salameh et al., 2018; Lichouri et al., 2018), where we added another feature which is the character-word_boundary (char_wb). In the following, we present a description of the three adopted features.

- **[Word n-grams:]** We extract n-gram word from 1st to 5th.
- **[Char n-grams:]** The character 1st to 5th grams are used as features.
- **[Char_wb n-grams:]** This feature creates character n-grams only from text inside word boundaries; n-grams at the edges of words are padded with space.

3.1.2 Additional Features

The features considered are obtained by applying three processes, either simultaneously or individually. These process are: Punctuation removable, Stop-word filtering, Normalization Process, Stemmer Process and a PosTagger Process. To deal with the last three process, we first defined our own normalizer function, then used the ISRIStemmer NLTK tool[1] for the second, whereas for last we used the NLTK postagger[2].

[1]https://kite.com/python/docs/nltk.stem.ISRIStemmer
[2]https://www.nltk.org/_modules/nltk/tag.htm

3.2 Proposed Approches

3.2.1 Lexical Approach

This approach is based on a set of text distance measures from the textdistance tools[3]. From a set of measures proposed by this tools we opted to choose one measure per category, namely: Hamming Distance, Mlipns Distance, Levenshtein Distance, Damerau Levenshtein Distance, Jaro Distance, Strcmp95 Distance, Needleman Wunsch Distance, Gotoh Distance, and the Smith Waterman Distance.

3.2.2 Statistical Approach

Based on a set of classifiers using the scikit-learn library (Pedregosa et al., 2011), namely: Linear Support Vector Classification (LSVC), Bernoulli Naive Bayes (BNB), Multinomial Naive Bayes (MNB), Logistic Regression (LGR), Stochastic Gradient Descent (SGD), Perceptron (PRP) and the Passive Aggressive (PAG), a statistical approach was proposed. Where we will consider the semantically similarity between questions as a binary classification problem with two classes: similar (1) or non similar (0).

3.2.3 Neural Approach

In this approach, we will consider a Text Classification Methods using a Siamese Deep Neural Network [4]. While using this script, we adopted for multiple configurations by varying the default setup: EMBEDDING_DIM = 50, MAX_SEQUENCE_LENGTH = 10, VALIDATION_SPLIT = 0.1, RATE_DROP_LSTM = 0.17, RATE_DROP_DENSE = 0.25, NUMBER_LSTM = 50, NUMBER_DENSE_UNITS = 50, ACTIVATION_FUNCTION = 'relu'.

4 Results

As shown in figure 1, the first step to be conducted is to experiment with the three approaches in parallel. For the first approach, which is the lexical approach, we conducted a similarity measure study, by calculating the distance between the two questions by using several metrics while considering a range of threshold values between 10% to 100%. The best results are presented in table 2.

The three best results obtained by this approach are by the following measures: Smith Water-

[3]https://pypi.org/project/textdistance/
[4]https://github.com/amansrivastava17/lstm-siamese-text-similarity

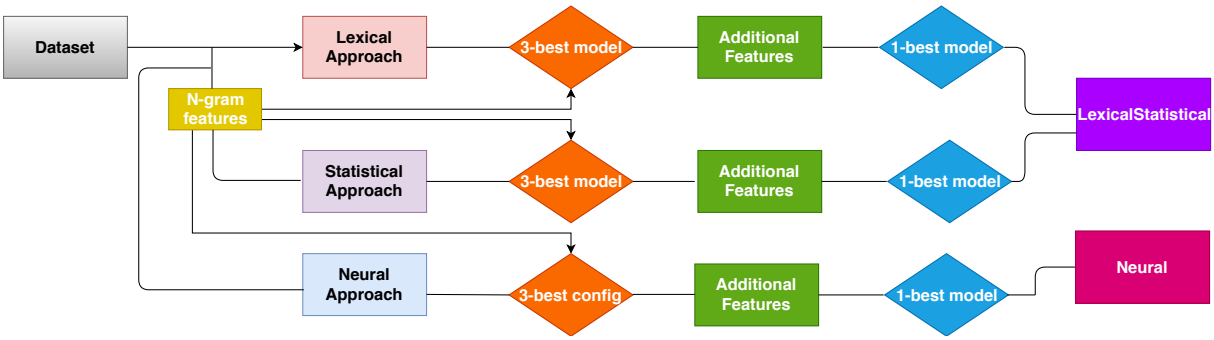


Figure 1: A Semantic Question Similarity System for Arabic Language

Similarity Measures	Threshold (%)	Score (%)
Hamming Distance	90	47.95
Mlipns Distance	90	38.82
Levenshtein Distance	40	**68.28**
Damerau Levenshtein Distance	40	67.91
Jaro Distance	30	64.20
Strcmp95 Distance	30	63.27
Needleman Wunsch Distance	50	**67.93**
Gotoh Distance	40	67.91
Smith Waterman Distance	30	**69.77**

Table 2: Best results obtained by the different measures while varying the threshold.

	MNB	BNB	LSVC	LGR	PRP	PAG	SGD
Unigram	63.11	63.98	70.86	69.70	66.43	68.81	70.87
Bigram	63.35	62.64	72.79	72.01	70.51	72.04	73.48
Trigram	62.41	60.46	73.94	72.55	69.75	72.50	73.98
4-grams	61.57	57.99	**74.25**	72.96	71.04	73.07	74.03
5-grams	60.70	56.34	74.11	73.11	71.17	**73.36**	**74.73**

Table 3: Results obtained by the used classifiers in term of F1-score while varying the number of grams n with the word feature

man distance, Levenshtein distance and Needleman Wunsch Distance. The best score obtained is around 69.77% with a threshold of 30. For the second approach, the statistical one, we used the n-grams word and char features, with a range of n from 1 to 5. The results obtained while applying these features with the aforementioned classifiers are presented in tables 3 and 4.

It should be noted, that with this approach; the three best results were obtained by the LSVC,

	MNB	BNB	LSVC	LGR	PRP	PAG	SGD
Unigram	44.48	64.72	67.60	66.59	57.57	54.60	66.38
Bigram	61.70	66.45	71.39	69.76	63.98	64.79	69.77
Trigram	**63.18**	**66.47**	73.03	70.98	68.13	66.86	71.28
4-grams	62.30	65.97	73.43	70.94	**69.51**	69.84	72.51
5-grams	61.68	65.37	**74.03**	**71.06**	69.48	**71.70**	**72.95**

Table 4: Results obtained by the used classifiers in term of F1-score while varying the number of grams n with the char feature

	max_length	function	threshold	F1-Score
config1	10	softmax	0.5	79.89
config2	10	softmax	0.55	79.89
config3	15	softmax	0.55	79.11

Table 5: Results obtained by the three best configurations in term of F1-score while varying the different parameters

		Development Score	
Approach	**Models**	**Public (30%)**	**Private (70%)**
Lexical	***NW***	70.37	68.08
	Lv	69.12	68.35
	SW	72.26	69.78
Statistical	***LSVC***	75.31	75.62
	SGD	75.13	74.77
	PAG	74.59	73.58
Neural	***Config1***	81.23	80.54
	Config2	82.31	79.77
	Config3	82.58	82.58

Table 6: Development results obtained by the three best models for each approach in the first step.

PAG and SGD while using the word (4/5)-grams feature. The f1-score obtained range between $\tilde{7}2\%$ and $\tilde{7}5\%$.

Whereas for third approach, based on the Siamese DNN and as mentioned in the description of the neural approach, we have experimented with multiple combination of values for each parameters and thus noted the three best configurations, which we presented in table 5. We can note that there is a net amelioration of the F1-score against the two previous approach with an amelioration of +5 point.

Before presenting the results that we obtained in the second step, we will report the development accuracy that we obtained after submitting the three best model for the three approach to the kaggle shared task website in table 6.

From the table 6, we can see that the best results

	Baseline	Features							
		P	**S**	**N**	**St**	**Pos**	**PSNPos**	**PSNSt**	**PSNStPos**
SW	69.77(30)	69.86(30)	70.62(30)	69.86(30)	69.77(30)	68.71(30)	**71.83(30)**	71.23(30)	**71.83(30)**
Lv	68.28(40)	68.46(40)	69.50(30)	68.17(40)	68.28(40)	66.90(30)	68.36(30)	70.65(40)	68.36(30)
NW	67.91(50)	68.30(50)	69.15(50)	67.93(50)	67.91(50)	66.78(30)	68.04(50)	70.01(50)	68.04(50)
LSVC(4)	74.25	73.94	69.58	72.86	73.79	**75.69**	74.16	69.60	74.37
SGD(5)	74.73	74.31	69.05	74.13	74.1	74.49	74.92	68.65	74.49
PAG(5)	73.36	72.84	68.62	74.01	72.96	73.53	73.37	68.02	73.28
DNN	79.89	77.29	75.65	76.66	74.76	**80.55**	74.46	74.66	75.82

Table 7: Comparison of the impact of the preprocessing step on the results obtained by the best models in the baseline system, in accordance to the three proposed approach. For the first parts of the table(lexical approach), we noted the threshold that gave us the best results in brackets.

			Development Score	
Approach	**Models**	**Features**	**Public (30%)**	**Private (70%)**
Lexical	**SmithWater**	**PNSStPos**	73.16	71.55
Statistical	***LSVC***	**PosTagger**	81.68	79.04
Neural	***Config1***	**PosTagger**	59.97	62.11

Table 8: Development results obtained by the 1-best models for each approach in the second step.

	Development Score	
Approach	**Public (30%)**	**Private (70%)**
1st Best Model	83.57	82.69
2nd Best Model	82.58	80.50
Benchmark	71.99	71.43

Table 9: Comparison of our best model performance against the benchmark.

are obtained by the neural approach in both test dataset (30% and 70%) with an average f1-score of more than 80%.

We will now, present the results obtained in the second step, where we applied a set of additional features. This step will permit us to select the best model for each approach. The table 7, present the gotten results.

By applying some additional features namely: Punctuation removal, Stopwords filter, Normalizer process, Stemmer process and PosTagger process, individually or sequentially, we can note a net amelioration of results for all the three approach by $\tilde{+}2$, $\tilde{+}2$ and $\tilde{+}1$ for the lexical, statistical and neural approaches, respectively.

When looking at the table 7, we can infer that the best model for each approach is as follows: the Smith Waterman distance for the lexical approach, the LSVC classifier for the statistical approach and the DNN+Postagger for the neural approach.

As we did before, we have re-submitted the best model for each approach to the kaggle to have the score with the test dataset. The gotten results are demonstrated in table 8.

We can note that despite the neural approach has scored the best score of 80.55% in the training phase, it could not well generalize on the test data, where it yielded 59.97% and 62.11% for both the public and private set. For the third step, we have compared the performance of a combination between the lexical and statistical approaches against the neural approach, which have given us two model: lexical+statistical and neural.

We started with the statistical approach, where we have opted to add a combination of features, which has given rise to a new features. This new feature contains:

- A 5-grams word feature.
- A 3-grams char feature.
- A 3-grams char_wb feature.

After that we used a TFidf transformation on the resulted matrix, which we will call **tf_mat1**.

For the lexical approach, we converted the resulted distance measures between the question pairs to an array, which we will call **dist_fea**.

Afer that we combined these two matrix **tf_mat1** and **dist_fea**, which we will call **tf_train**, that will be used as input to the LSVC classifier.

This combination has permitted us to have our best performance in this shared task with an average score of 83.13%. Whereas the neural approach has given use our 2nd best model with an average score of 81.50%. Table 9 present a comparison of our two best models against the benchmark.

5 Conclusion

In this paper, we presented *ST NSURL 2019 Shared Task: Semantic Question Similarity in Arabic* that participated in the *2019 NSURL Shared*

Task 8 (Semantic Question Similarity in Arabic). The performance of our best run on the test data for this Task ranked 7 between 9 teams in both private and public data sets. In this approach, we used a Linear Support Vector classifier by utilizing a combination of word, char and char_wb n-gram as features as well as a lexical approach-based model (Smith-Waterman), plus a PosTagger process.

Despite the simplicity of these features, we got promising results which encourage us to do further experiments on other features such us LSA that may lead to better results.

References

Lahsen Abouenour, Karim Bouzouba, and Paolo Rosso. 2010. An evaluated semantic query expansion and structure-based approach for enhancing arabic question/answering. *International Journal on Information and Communication Technologies*, 3(3):37–51.

Yassine Benajiba, Paolo Rosso, and Abdelouahid Lyhyaoui. 2007. Implementation of the arabiqa question answering system's components. In *Proc. Workshop on Arabic Natural Language Processing, 2nd Information Communication Technologies Int. Symposium, ICTIS-2007, Fez, Morroco, April*, pages 3–5.

W Brini, M Ellouze, and L Hadrich Belguith. 2009. Qasal: Un système de question-réponse dédié pour les questions factuelles en langue arabe. *9ème Journées Scientifiques des Jeunes Chercheurs en Génie Electrique et Informatique, Tunisia*.

Davide Buscaldi, Paolo Rosso, José Manuel Gómez-Soriano, and Emilio Sanchis. 2010. Answering questions with an n-gram based passage retrieval engine. *Journal of Intelligent Information Systems*, 34(2):113–134.

Bassam Hammo, Hani Abu-Salem, and Steven Lytinen. 2002. Qarab: A question answering system to support the arabic language. In *Proceedings of the ACL-02 workshop on Computational approaches to semitic languages*, pages 1–11. Association for Computational Linguistics.

Mohamed Lichouri, Mourad Abbas, Abed Alhakim Freihat, and Dhiya El Hak Megtouf. 2018. Word-level vs sentence-level language identification: Application to algerian and arabic dialects. *Procedia Computer Science*, 142:246–253.

FA Mohammed, Khaled Nasser, and HM Harb. 1993. A knowledge based arabic question answering system (aqas). *ACM SIGART Bulletin*, 4(4):21–30.

Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016. SemEval-2016 task 3: Community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California. Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Mohammad Salameh, Houda Bouamor, and Nizar Habash. 2018. Fine-grained arabic dialect identification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1332–1344.

Haitham Seelawi, Ahmad Mustafa, Al-Bataineh Hesham, Wael Farhan, and Hussein T. Al-Natsheh. 2019. NSURL-2019 task 8: Semantic question similarity in arabic. In *Proceedings of the first International Workshop on NLP Solutions for Under Resourced Languages*, NSURL '19, Trento, Italy.

Statistical Machine Translation between Myanmar (Burmese) and Dawei (Tavoyan)

Thazin Myint Oo
UCSY, Myanmar
thazinmyintoo@ucsy.edu.mm

Ye Kyaw Thu
NECTEC, Thailand
ka2pluskha2@gmail.com

Khin Mar Soe
UCSY, Myanmar
khinmarsoe@ucsy.edu.mm

Thepchai Supnithi
NECTEC, Thailand
thepchai.supnithi@nectec.or.th

Abstract

This paper contributes the first evaluation of the quality of statistical machine translation (SMT) between Myanmar (Burmese) and Dawei (Tavoyan). We also developed a Myanmar-Dawei parallel corpus (around 9K sentences) based on the Myanmar language of ASEAN MT corpus. The 10 folds cross-validation experiments were carried out using three different statistical machine translation approaches: phrase-based, hierarchical phrase-based, and the operation sequence model (OSM). In addition, two types of segmentation were studied: word and syllable segmentation. The results show that all three statistical machine translation approaches give comparable BLEU and RIBES scores for both Myanmar to Dawei and Dawei to Myanmar machine translations. OSM approach achieved the highest BLEU and RIBES scores among three SMT approaches for both word and syllable segmentation.

1 Introduction

Our main motivation for this research is to investigate SMT performance for Myanmar (Burmese) and Dawei (Tavoyan) language pair. The Dawei (Tavoyan) language is closely related to Myanmar (Burmese) language and it is often considered as dialect of Myanmar language. The state-of-the-art techniques of statistical machine translation (SMT) (Koehn et al., 2003). This demonstrate good performance on translation of languages with relatively similar word orders (Koehn, 2005). To date, there have been some studies on the SMT of Myanmar language. (Thu et al., 2016) presented the first large-scale study of the translation of the Myanmar language. A total of 40 language pairs were used in the study that included languages both similar and fundamentally different from Myanmar. The results show that the hierarchical phrase-based SMT (HPBSMT) (Chiang, 2007) approach gave the highest translation quality in terms of both the BLEU (Papineni et al., 2002) and RIBES scores (Isozaki et al., 2010). Win Pa Pa et al (2016) (Pa et al., 2016) presented the first comparative study of five major machine translation approaches applied to low-resource languages. Phrase-based statistical machine translation (PBSMT), HPBSMT, tree-to-string (T2S), string-to-tree (S2T) and operation sequence model (OSM) translation methods to the translation of limited quantities of travel domain data between English and Thai, Laos, Myanmar in both directions. The experimental results indicate that in terms of adequacy (as measured by BLEU score), the PBSMT approach produced the highest quality translations. Here, the annotated tree is used only for English language for S2T and T2S experiments. This is because there is no publicly available tree parser for Lao, Myanmar and Thai languages. According to our knowledge, there is no publicly available tree parser for both Dawei and Myanmar languages and thus we cannot apply S2T and T2S approaches for Myanmar-Dawei language pair. From their RIBES scores, we noticed that OSM approach achieved best machine translation performance for Myanmar to English translation. Moreover, we learned that OSM approach gave highest translation performance translation between Khmer (the official language of Cambodia) and twenty other languages, in both directions (Thu et al., 2015). Relating to Myanmar langauge dialects, Thazin Myint Oo et al. (2018) (Oo et al., 2018) contributed the first PBSMT, HPBSMT and OSM machine translation evaluations between Myanmar and Rakhine. The experiment was used the 18K Myanmar-Rakhine parallel cor-

pus that constructed to analyze the behavior of a dialectal Myanmar-Rakhine machine translation. The results showed that higher BLEU (57.88 for Myanmar-Rakhine and 60.86 for Rakhine-Myanmar) and RIBES (0.9085 for Myanmar-Rakhine and 0.9239 for Rakhine-Myanmar) scores can be achieved for Rakhine-Myanmar language pair even with the limited data. Based on the experimental results of previous works, in this paper, the machine translation experiments between Myanmar and Dawei were carried out using PBSMT, HPBSMT and OSM.

2 Related Work

Karima Meftouh et al. built PADIC (Parallel Arabic Dialect Corpus) corpus from scratch, then conducted experiments on cross dialect Arabic machine translation (Meftouh et al., 2015) PADIC is composed of dialects from both the Maghreb and the Middle-East. Some interesting results were achieved even with the limited corpora of 6,400 parallel sentences. Using SMT for dialectal varieties usually suffers from data sparsity, but combining word-level and character-level models can yield good results even with small training data by exploiting the relative proximity between the two varieties (Neubarth et al., 2016). Friedrich Neubarth et al. described a specific problem and its solution, arising with the translation between standard Austrian German and Viennese dialect. They used hybrid approach of rule-based preprocessing and PBSMT for getting better performance. Pierre-Edouard Honnet et al. proposed solutions for the machine translation of a family of dialects, Swiss German, for which parallel corpora are scarce (Honnet et al., 2018). They presented three strategies for normalizing Swiss German input in order to address the regional and spelling diversity. The results show that character-based neural MT was the most promising one for text normalization and that in combination with PBSMT achieved 36 % BLEU score.

3 Dawei Language

The Tavoyan or Dawei dialect of Burmese is spoken in Dawei (Tavoy), in the coastal Tanintharyi Region of southern Myanmar (Burma). The large and quite distinct Dawei or Tavoyan variety is spoken in and around Dawei (formerly Tavoy) in Tanintharyi (formerly Tenasserim) by about 400,000 people; its sterotyped characteristic is the mesial /I/, found in earlist Bagan inscriptions but by merger there nearly 800 years ago; for further information see Pe Maung Tin (1933) and Okell (1995)(OKELL, 1995). Dawei is a city of south-eastern Myanmar and is the capital of Tanintharyi Region, formerly known as the Tenasserim is bounded by Mon state to the north, Thailand to the east and south, and the Andaman sea to the west. Tavoyan retains /-l-/ medial that has since merged into the /-j-/ medial in standard Burmese and can form the following consonant clusters: /gl-/, /kl-/, /kʰl-/, /bl-/, /pl-/, /pʰl-/, /ml-/, /m̥l-/. Examples include "ဧမ္လ" (/mlè/ → Standard Burmese /mjè/) for "ground" and "ကျောင်း" (kláʊɴ/ → Standard Burmese tʃáʊɴ/) for "school". [4] Also, voicing only with unaspirated consonants, whereas in standard Burmese, voicing can occur with both aspirated and unaspirated consonants. Also, there are many loan words from Malay and Thai not found in Standard Burmese. An example is the word for goat, which is hseit "ဆိတ်" in Standard Burmese but be "ဘဲ" in Tavoyan. In the Tavoyan dialect, terms of endearment, as well as family terms, are considerably different from Standard Burmese. For instance, the terms for "son" and "daughter" are "ဖစု့" (/pʰa̰ òu/) and "မိစု့"(/mḭ òu/) respectively. Moreover, the honorific "နောင်" (Naung) is used in lieu of "မောင်" (Maung) for young males. Another evidence of "Dawei" is "Dhommarazaka" pogoda inscription of Bagan period. It was inscription of Bagan period. It was inscribed in AD 1196 during the region of Bagan King Narapatisithu (AD 1174-1201) . In this inscription line 6 to 19, when the demarcation of Bagan is mentioned "Taung-Kar-Htawei" (up to Htawei to the south) and "Taninthaye" (Tanintharyi) are including. Therefore, the name of "Dawei" appeared particulary since Bagan period, at the time of the first Myanmar Empire. (Dawei was established at Myanmar year 1116) is actually meant that the present name Dawei appears as the name of the settlers later and the original name of the city is Tharyarwady, which was established at Myanmar year 1116 according to the saying. As "Dawei" nationality deserves as one nationalist in our country. Actually, Dawei region is a place where local people lived since very ancient Stone

Age. After that, Stone Age, Bronze Age and Iron Age culture developed. Moreover, as there has sound evidence of Thargara ancient city, comtemporary to Phu Period, the Dawei people, can be assumed that they are one nationality of high culture in Myanmar. Dawei(Tavoyan) usage and vocabularies is divided into three main groups. The first one is using Myanmar vocabularies with Dawei speech, the second is the vocabularies same with Myanmar vocabularies and using isolated Dawei words and vocabularies. In Myanmar word (“ထို, ဟို”), (“here, there”) is used “သယ်” (“here”) and “ဟောက်” (“there”) in Dawei language. For example Dawei word “သယ်မျိုး” is same as “ဒီလို” in Myanmar language and “ဟောက်မျိုး” means “ဟိုလို” in Myanmar language. The question words “နည်း (သနည်း), လဲ (သလဲ)” are used in Myanmar language, similarly “လော,လော်” is used instead of “လား (သလား)” in Dawei language. Moreover, “ဘာလဲ”(what) and “ဘာဖြစ်တာလဲ” (“what happened”) is same with “ဖြာန္ဒး” and “ဖြာဖြစ်န္ဒး” in Dawei usage. In negative sense of Myanmar word “ဘူး” is not usually used in Dawei word. The negative Dawei words are “ဟ့ (ရ)” or “ဟန်း” (“No” in English). Myanmar adverb word “သိပ်, အလွန်, အလွန်အလွန်”(very, extremely) is used as “ရရာ, ရမိရရာ, ဗြင်း”. Some more example of Dawei vocabularies are “ဝန်းရှင်း” (“ကိုယ်ဝန်ဆောင်” in Myanmar language, “pregnant” in English), “ကောန်သား” (“ကောင်လေး” in Myanmar language, “boy” in English), “ဝယ်သား” (“ကောင်မလေး” in Myanmar language, “girl” in English), “ကပ်” (“ပိုက်ဆံ” in Myanmar language, “money” in English), “ချော့-က်တိုအိုးသီး” (“ကွဲကောသီး” in Myanmar language, “pomelo” in English) and “သစ်ခတ်ကွား” (“ကျားသစ်” in Myanmar language, “leopard” in English). The followings are some example parallel sentences of Myanmar (my) and Dawei(dw):

dw: သယ်ဝယ်သား က လှ ဗြင်း ဟ့ယ် ။
my: ဒီကောင်မလေး က လှ လွန်း တယ် ။
(“The girl is so beautiful” in English)

dw: လတ်ဖတ်ရယ် က ရှိ ဗြင်း ဟ့ယ် ။
my: လက်ဖက်ရည် က ချို လွန်း တယ် ။
(“The tea is so sweet” in English)

dw: ကောန်သား ကျောန်း မှန်းမှန် သွား ဟ့ယ် ။
my: ကောင်လေး ကျောင်း မှန်မှန် တက် တယ် ။
(“The boy goes to school regularly” in English)

4 Methodology

In this section, we describe the methodology used in the machine translation experiments for this paper.

4.1 Phrase-Based Statistical Machine Translation

A PBSMT translation model is based on phrasal units (Koehn et al., 2003). Here, a phrase is simply a contiguous sequence of words and generally, not a linguistically motivated phrase. A phrase-based translation model typically gives better translation performance than word-based models. We can describe a simple phrase-based translation model consisting of phrase-pair probabilities extracted from corpus and a basic reordering model, and an algorithm to extract the phrases to build a phrase-table (Specia, 2011). The phrase translation model is based on noisy channel model. To find best translation $\hat{e}$ that maximizes the translation probability $\mathbf{P}(f)$ given the source sentences; mathematically. Here, the source language is French and the target language is an English. The translation of a French sentence into an English sentence is modeled as equation 1.

$$\hat{e} = argmax_e \mathbf{P}(e|f) \quad (1)$$

Applying the Bayes' rule, we can factorized the into three parts.

$$P(e|f) = \frac{\mathbf{P}(e)}{\mathbf{P}(f)} \mathbf{P}(f|e) \quad (2)$$

The final mathematical formulation of phrase-based model is as follows:

$$argmax_e \mathbf{P}(e|f) = argmax_e \mathbf{P}(f|e)\mathbf{P}(e) \quad (3)$$

We note that denominator $\mathbf{P}(f)$ can be dropped because for all translations the probability of the source sentence remains the same . The $\mathbf{P}(e|f)$ variable can be viewed as the bilingual dictionary with probabilities attached to each entry to the dictionary (phrase table). The $\mathbf{P}(e)$ variable governs the grammaticality of the translation and we model it using n-gram language model under the PBMT paradigm.

[X] [X] ကို တောပန် [X] ||| [X] [X] ကို တောင်းပန် [X]
[X] [X] ကို တွန်းပန် [X] ||| [X] [X] မ တောင်းပန် [X]
[X] [X] ကို တွန်းပန် ဟုလား [X] ||| [X] [X] မ တောင်းပန် ဘူးလား [X]
[X] [X] ကို တို့ [X] ||| [X] [X] ကို တွေ့ [X]
[X] [X] ကို တို့ ဟို့ [X] ||| [X] [X] ကို တွေ့ ဖို့ [X]

Figure 1: Some examples of hierarchical phrase-based grammar between Dawei and Myanmar phrases

4.2 Hierarchical Phrase-Based Statistical Machine Translation

The hierarchical phrase-based SMT approach is a model based on synchronous context-free grammar (Specia, 2011). The model is able to be learned from a corpus of unannotated parallel text. The advantage this technique offers over the phrase-based approach is that the hierarchical structure is able to represent the word re-ordering process. The re-ordering is represented explicitly rather than encoded into a lexicalized re-ordering model (commonly used in purely phrase-based approaches). This makes the approach particularly applicable to language pairs that require long-distance re-ordering during the translation process (Braune et al., 2012). Some examples of hierarchical phrase based grammar between Dawei and Myanmar phrases are shown in Figure 1.

4.3 Operation Sequence Model

The operation sequence model that can combines the benefits of two state-of-the-art SMT frameworks named n-gram-based SMT and phrase-based SMT. This model simultaneously generate source and target units and does not have spurious ambiguity that is based on minimal translation units (Durrani et al., 2011) (Durrani et al., 2015). It is a bilingual language model that also integrates reordering information. OSM motivates better reordering mechanism that uniformly handles local and non-local reordering and strong coupling of lexical generation and reordering. It means that OSM can handle both short and long distance reordering. The operation types are such as generate, insert gap, jump back and jump forward which perform the actual reordering. The following shows an example translation process of English sentence "Please sit here" into Myanmar language with the OSM.

Source: Please sit here
Target: ကျေးဇူးပြုပြီး ဒီမှာ ထိုင်

Operation 1: Generate (Please, ကျေးဇူးပြုပြီး)

Operation 2: Insert Gap

Operation 3: Generate (here, ကျေးဇူးပြုပြီး ဒီမှာ)

Operation 4: Jump Back (1)

Operation 5: Generate (sit, ကျေးဇူးပြုပြီး ဒီမှာ ထိုင်)

5 Experiment

5.1 Corpus Statistics

We used 9,000 Myanmar sentences (without name entity tags) of the ASEAN-MT Parallel Corpus (Prachya and Thepchai, 2013), which is a parallel corpus in the travel domain. It contains six main categories and they are people (greeting, introduction and communication), survival (transportation, accommodation and finance), food (food, beverage and restaurant), fun (recreation, traveling, shopping and nightlife), resource (number, time and accuracy), special needs (emergency and health). Manual Translation into Rakhine Language was done by native Rakhine students from two Myanmar universities and the translated corpus was checked by the editor of Rakhine newspaper. Word segmentation for Rakhine was done manually. We held 10-fold cross-validation experiments and used 6,883 to 6,893 sentences for training, 1,212 to 1,217 sentences for development and 890 to 922 sentences for evaluation respectively.

5.2 Word Segmentation

In both Myanmar and Dawei text, spaces are used for separating phrases for easier reading. It is not strictly necessary, and these spaces are rarely used in short sentences. There are no clear rules for using spaces, and thus spaces may (or may not) be inserted between words, phrases, and even between a root words and their affixes. Although Myanmar sentences of ASEAN-MT corpus is already segmented, we have to consider some rules for manual word segmentation of

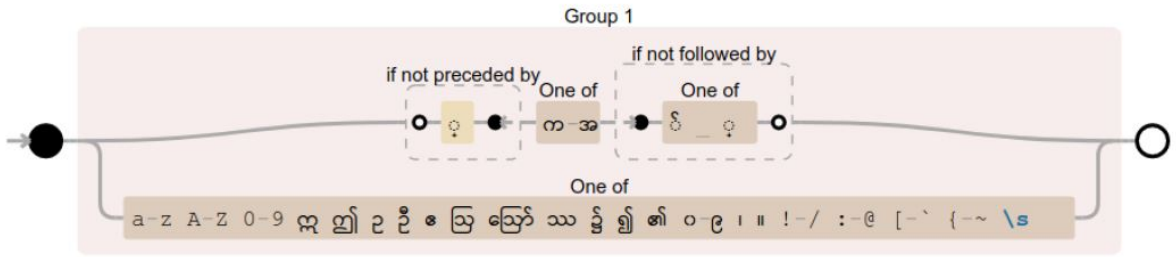


Figure 2: Visualizaiton of syllable breaking with regular expression for Myanmar language

Dawei sentences. We defined Dawei "word" to be meaningful units and affix, root word and suffixe(s) are separated such as "စား ဟုယ်", "စားပီးဟုယ်", "စား ဖို့ဟုယ်". Here, "စား" ("eat"in English) is a root word and the others are suffixes for past and future tenses. Similar to Myanmar language, Dawei plural nouns are identified by following particle. We also put a space between noun and the following particle, for example a Dawei word "ဖွန်သားဒေ" (shrimp) is segmented as two words "ဖွန်သား" and the particle "ဒေ". In Dawei grammar, particles describe the type of noun, and used after number or text number. For example, a Dawei word "ရှိုးခိုသီးတစ်လုံး" ("papaya" in English) is segmented as "ရှိုးခိုသီး တစ် လုံး". In our manual word segmentation rules, compound nouns are considered as one word and thus, a Dawei compound word "ကပ်" + "အိတ်" ("money" + "bag" in English) is written as one word "ကပ်အိတ်" ("wallet" in English). Dawei adverb words such as "ရရာ, ရမိရရာ" ("very" in English), "ပြင်း" ("extremely" in English) are also considered as one word. The following is an example of word segmentation for a Dawei sentence in our corpus and the meaning is "Shrimps are very rare and bought fishes."

Unsegmented Dawei sentence:
dw: ဖွန်သားဒေရရာရှားဟုယ်၊ငါးဗောင်းသားဘဲ့ဝယ်လာရဟုယ်။

Word Segmented Dawei sentence:
dw: ဖွန်သား ဒေ ရရာ ရှား ဟုယ် ၊ ငါးဗောင်းသား ဘဲ့ ဝယ် လာရဟုယ် ။

In this example, "ဖွန်သားဒေ" (shrimps) is segmented as two words "ဖွန်သား" and the particle "ဒေ". Dawei adverb words such as "ရရာ" ("rare" in English) is also considered as one word and a root word "ဝယ်" and the suffix "လာရဟုယ်" are also segmented as two words "ဝယ် လာရဟုယ်" ("bought" in English)

5.3 Syllable Segmentation

Generally, Myanmar words are composed of multiple syllables, and most of the syllables are composed of more than one character. Syllables are composed of Myanmar words. If we only focus on consonant-based syllables, the structure of the syllable can be described with Backus normal form (BNF) as follows:

$$\text{Syllable} := \text{CMW[CK][D]}$$

Here, C stands for consonants, M for medials, V for vowel, K for vowel killer character, and D for diacritic characters. Myanmar syllable segmentation can be done with a rule-based approach, finite state automation (FSA) or regular expressions (RE) (https://github.com/ye-kyawthu/sylbreak). The visualization of the syllable breaking based on the RE for Myanmar language is as shown in Figure 2. In our experiments, we used RE based Myanmar syllable segmentation tool named "sylbreak". The following is an example of syllable segmentation for a Dawei sentence in our corpus and the meaning is "You are cute."

Unsegmented Dawei sentence:
dw: နန်ရှစ်ဇရာကွန်းဇမား။

Syllable segmented Dawei sentence:
dw: နန် ရှစ် ဇ ရာ ကွန်း ဇ မား ။

5.4 Moses SMT System

We used the PBSMT, HPBSMT and OSM system provided by the Moses toolkit (Koehn et al., 2007) for training the PBSMT, HPBSMT and OSM statistical machine translation systems. The word segmented source language was aligned with the word segmented target language using GIZA++

src-tgt	PBSMT	HPBSMT	OSM
dw-my	29.143 (0.82286)	29.09 (0.82203)	**29.563 (0.82369)**
my-dw	21.575(0.62624)	21.697 (0.78651)	**21.701 (0.78667)**

Table 1: Average BLEU and RIBES scores for PBSMT, HPBSMT and OSM using word segmentation

src-tgt	PBSMT	HPBSMT	OSM
dw-my	60.788 (0.94613)	60.472 (0.94476)	**63.221 (0.94825)**
my-dw	44.8 (0.91601)	45.441 (0.91496)	**45.584 (0.91550)**

Table 2: Average BLEU and RIBES scores for PBSMT, HPBSMT and OSM using Syllable Segmentation

(Och and Ney, 2000). The alignment was symmetrize by grow-diag-final and heuristic [1]. The lexicalized reordering model was trained with the msd-bidirectional-fe option (Tillmann, 2004). We use KenLM (Heafield, 2011) for training the 5-gram language model with modified Kneser-Ney discounting (Chen and Goodman, 1996). Minimum error rate training (MERT) (Och, 2003) was used to tune the decoder parameters and the decoding was done using the Moses decoder (version 2.1.1). We used default settings of Moses for all experiments.

6 Evaluation

We used two automatic criteria for the evaluation of the machine translation output. One was the de facto standard automatic evaluation metric Bilingual Evaluation Understudy (BLEU) (Papineni et al., 2002) and the other was the Rank-based Intuitive Bilingual Evaluation Measure (RIBES) (Isozaki et al., 2010). The BLEU score measures the precision of n-gram (over all n $\leq$ 4 in our case) with respect to a reference translation with a penalty for short translations (Papineni et al., 2002). Intuitively, the BLEU score measures the adequacy of the translation and large BLEU scores are better. RIBES is an automatic evaluation metric based on rank correlation coefficients modified with precision and special care is paid to word order of the translation results. The RIBES score is suitable for distance language pairs such as Myanmar and English. Large RIBES scores are better.

7 Results and Discussion

The BLEU and RIBES score results for machine translation experiments with PBSMT, HPBSMT and OSM are shown in Table 1. Bold numbers indicate the highest scores among three SMT approaches. The RIBES scores are inside the round brackets. Here, "my" stands for Myanmar, "dw" stands for Dawei, "src" stands for source language and "tgt" stands for target language respectively. The BLEU and RIBES score results for machine translation experiments with PBSMT, HPBSMT and OSM using word level segmentation between Myanmar and Dawei languages are shown in Table 1. From the results, OSM method achieved the highest BLEU and RIBES score for both Myanmar-Dawei and Dawei-Myanmar machine translations. Interestingly, the BLEU and RIBES score of all three methods are comparable performance. Our results with current parallel corpus indicate that Dawei to Myanmar machine translation is better performance (around 8 BLEU and 0.03 RIBES scores higher) than Myanmar to Dawei translation direction. The results of BLEU and RIBES scores of syllable segmentaion between Myanmar and Dawei languages are shown in Table 2. Our results with syllable segmentation also indicate that Dawei to Myanmar machine translation is better performance (around 17 BLEU and 0.03 RIBES score higher) than Myanmar to Dawei translation direction. As we expected, generally, machine translation performance of all three SMT approaches between Myanmar and Dawei languages with limited parallel corpus achieved suitable scores for both BLEU and RIBES. The reason is that as we mentioned in Section 3, the two languages, Myanmar and Dawei are close languages. We assume that long distance reordering is relatively rare and only local reordering is enough for the Myanmar-Dawei language pair. We can expect that we can increase these scores higher than current results by increasing the corpus size in the near future.

8 Error Analysis

We also used the SCLITE (score speech recognition system output) program from the NIST scor-

Freq	Reference ==> Hypothesis
16	သူမ ==> သူ
14	ခင်ဗျား ==> မင်း
9	ပါတယ် ==> တယ်
8	ပါဘူး ==> ဘူး
7	သလဲ ==> တယ်
5	ဘာတွေ ==> ဘာ
5	မင်းကို ==> ကို
5	မလား ==> မှာလား
5	လား ==> သလား
5	အဲဒါကို ==> ကို
4	ခဲ့ဘူး ==> ဘူး
4	ဘူးလား ==> ရှိလား
4	မင်းရဲ့ ==> မင်း
4	လဲ ==> သလဲ
4	သူ့ ==> သူမ

Table 3: The top 15 confusion pairs of OSM model for Dawei-Myanmar machine translation with word segmentation

ing toolkit SCTK version 2.4.10 for making dynamic programming based alignments between reference and hypothesis strings for detail analysis on translation errors. From our studies, the top 15 confusion matrix for Dawei-Myanmar OSM machine translation (with word segmentation) can be seen in Table 3. We also made manual error analysis on translated outputs of the best OSM model, and we found that dominant errors are different in sentence level. We will introduce four frequent error patterns and they are "Male-Female Vocabulary Error", "Paraphrasing Error", "Word Segmentation Error" and "Negative Error". The followings are some example translation mistakes for each category:

Male-Female Vocabulary Error

SOURCE: **သူ နန့် ဟို မြင် လား ။**
Scores: (#C #S #D #I) 3 2 0 1
REF: ****** သူမ မင်းကို မြင် သလား ။
HYP: သူ မင်း ကို မြင် သလား ။
Eval: I S S

SOURCE: **သူ့ကိုယ်သူ သိ ဟုယ် ။**
Scores: (#C #S #D #I) 3 1 0 0
REF: သူမကိုယ်သူမ သိ ပါတယ် ။
HYP: သူ့ကိုယ်သူ သိ ပါတယ် ။
Eval: S

Paraphrasing Error

SOURCE:**ဌား ဟုးဟို့ အီ လေ ။**
Scores: (#C #S #D #I) 4 1 0 0
REF: ဌားရမ်း ထားတဲ့ အိမ် တွေ ။
HYP: ဌား ထားတဲ့ အိမ် တွေ ။
Eval: S

SOURCE: **လူတိုင်း သတ္တိ ရှိ ကေ့ဟုယ် ။**
Scores: (#C #S #D #I) 4 1 0 0
REF: လူတိုင်း သတ္တိ ရှိ ကြပါတယ် ။
HYP: လူတိုင်း သတ္တိ ရှိ ကြတယ် ။
Eval: S

SOURCE: **ကျွန်တော် အိ ရှင်နေဟုယ် ။**
Scores: (#C #S #D #I) 3 1 0 2
REF: ကျွန်တော် အိပ် **** ****** ချင်နေတယ် ။
HYP: ကျွန်တော် အိပ် ဖို့ ဆန္ဒရှိ တယ် ။
Eval: I I S

SOURCE: **သူဟု ရတိုင်း လု မား ။**
Scores: (#C #S #D #I) 3 2 0 0
REF: သူက အရမ်း လု တာပဲ ။
HYP: သူက သိပ် လု ရော ။
Eval: S S

Word Segmentation Error

SOURCE: **အဲဝယ်ဟုး ကားမွန်း ဟို့မှုဝလား ။**
Scores: (#C #S #D #I) 4 1 1 0
REF: သူမ ကား မောင်း မှာ မဟုတ်ဘူးလား ။
HYP: သူမ ********* ကားမောင်း မှာ မဟုတ်ဘူးလား ။
Eval: D S
SOURCE: **အယ်မိဇာ ပိုဆိုး လာဟုယ် ။**
Scores: (#C #S #D #I) 3 1 1 0
REF: အဲဒါ ပို ဆိုး လာတယ် ။
HYP: အဲဒါ ********* ပိုဆိုး လာတယ် ။
Eval: D S

Negative Error

SOURCE: **ဖြေ ပေး ဟို့ ရှစ် နေလား ။**
Scores: (#C #S #D #I) 5 1 0 1
REF: အဖြေ *** ပေး ဖို့ ရှက် နေသလား ။
HYP: အဖြေ မ ပေး ဖို့ ရှက် နေတာလား ။
Eval: I S

SOURCE: **ဝယ်ရား နှုတ်ဆက် သွား ဟု ။**
Scores: (#C #S #D #I) 5 0 1 0
REF: သူမ နှုတ်ဆက် မ သွား ဘူး ။
HYP: သူမ နှုတ်ဆက် *** သွား ဘူး ။
Eval: D

Where "SOURCE" is the test sentence of Dawei language, "Scores" are operation scores of the Edit Distance (Miller et al., 2009), "C" is the number of correct words, "S" is the number of substitutions, "D" is the number of deletions, "I" is the number of insertions, "REF" for reference (i.e. Myanmar sentence), "HYP" for hypothesis and "Eval" is the ordered sequence of edit operations.

We found that translation error of male to female vocabulary and vice versa happen between Dawei-Myanmar translation such as "သူမ" ("she" in English) to "သူ" ("he" in English), "သူမကိုယ်သူမ" ("herself" in English) to "သူကိုယ်သူ" ("himself" in English). The second category, paraphrasing errors are really interesting and it is also proved that two language are similar. In our paraphrasing error examples, the meanings of all reference and hypothesis pairs are the same. Some errors are just the difference between the formal (polite form) and informal written form such as "ကြပါတယ်" (polite form of ending phrase "ကြတယ်" in Myanmar conversation) and "ကြတယ်". One of the possible reasons for the word segmentation errors is inconsistent word segmentation of human translators such as "ကားမောင်း" and "ကား မောင်း" ("drive a car" in English). We also found that one more frequent translation errors between Dawei-Myanmar and Myanmar-Dawei machine translation is changing into negative form (e.g. "အဖြေပေး" ("to answer" in English) and "အဖြေမပေး" ("no answer" in English).

9 Conclusion

This paper contributes the first PBSMT, HPBSMT and OSM machine translation evaluations from Myanmar to Dawei and Dawei to Myanmar. We used the 9K Myanmar-Dawei parallel corpus that we constructed to analyze the behavior of a dialectal Myanmar-Dawei machine translation. We also investigated two types of segmentation schemes (word segmentation and syllable segmentation). We showed that well-grounded BLEU and RIBES scores can be achieved for Dawei-Myanmar language pair even with the limited data. In the near future we plan to test PBSMT, HPBSMT and OSM models with other Myanmar dialect languages such as Myeik (Beik).

Acknowledgment

We would like to thank U Aung Myo (Leading Charge, Dawei Ethnic Organizing Committee, DEOC) for his advice especially on writing system of Dawei language with Myanmar characters. We are very greatful to Daw Thiri Hlaing (Lecturer, University of Computer Studies Dawei) for her leading the Myanmar-Dawei Translation Team. We would like to thank all students of Myanmar-Dawei translation team namely, Aung Myat Shein, Aung Paing, Aye Thiri Htun, Aye Thiri Mon, Htet Soe San, Ming Maung Hein, Nay Lin Htet, Thuzar Win Htet, Win Theingi Kyaw, Zin Bo Hein and Zin Wai for translation between Myanmar and Dawei sentences. Last but not least, we would like to thank Daw Khin Aye Than (Prorector, University of Computer Studies Dawei) for all the help and support during our stay at University of Computer Studies Dawei.

References

Fabienne Braune, Anita Ramm, and Alexander Fraser. 2012. Long-distance reordering during search for hierarchical phrase-based smt. In Proceedings of the Annual Conference of the European Association for Machine Translation (EAMT), pages 177--184.

Stanley F. Chen and Joshua Goodman. 1996. An empirical study of smoothing techniques for language modeling. In Proceedings of the 34th Annual Meeting on Association for Computational Linguistics, ACL '96, pages 310--318, Stroudsburg, PA, USA. Association for Computational Linguistics.

David Chiang. 2007. Hierarchical phrase-based translation. Computational Linguistics, 33(2):201--228.

Nadir Durrani, Helmut Schmid, and Alexander Fraser. 2011. A joint sequence translation model with integrated reordering. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 1045--1054, Portland, Oregon, USA. Association for Computational Linguistics.

Nadir Durrani, Helmut Schmid, Alexander Fraser, Philipp Koehn, and Hinrich Schütze. 2015. The operation sequence Model---Combining n-gram-based and phrase-based statistical machine translation. Computational Linguistics, 41(2):157--186.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In Proceedings of the Sixth Workshop on Statistical Machine Translation, pages 187--197, Edinburgh, Scotland. Association for Computational Linguistics.

Pierre-Edouard Honnet, Andrei Popescu-Belis, Claudiu Musat, and Michael Baeriswyl. 2018. Machine translation of low-resource spoken dialects: Strategies for normalizing swiss German. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan. European Language Resources Association (ELRA).

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 944--952, Cambridge, MA. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In The tenth Machine Translation Summit, pages 79--86, Phuket, Thailand. AAMT, AAMT.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, ACL '07, pages 177--180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, NAACL '03, pages 48--54, Stroudsburg, PA, USA. Association for Computational Linguistics.

Karima Meftouh, Salima Harrat, Salma Jamoussi, Mourad Abbas, and Kamel Smaili. 2015. Machine translation experiments on PADIC: A parallel Arabic DIalect corpus. In Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation, pages 26--34, Shanghai, China.

Frederic P. Miller, Agnes F. Vandome, and John McBrewster. 2009. Levenshtein Distance: Information Theory, Computer Science, String (Computer Science), String Metric, Damerau Levenshtein Distance, Spell Checker, Hamming Distance. Alpha Press.

Friedrich Neubarth, Barry Haddow, Adolfo Huerta, and Harald Trost. 2016. A hybrid approach to statistical machine translation between standard and dialectal varieties. volume 9561, pages 341--353.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1, ACL '03, pages 160--167, Stroudsburg, PA, USA. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, ACL '00, pages 440--447, Stroudsburg, PA, USA. Association for Computational Linguistics.

John OKELL. 1995. Three burmese dialects. Papers in Southeast Asian Linguistics No.13, Studies in Burmese Languages, 13:1--138.

Thazin Myint Oo, Ye Kyaw Thu, and Khin Mar Soe. 2018. Statistical machine translation between myanmar (burmese) and rakhine (arakanese). In Proceedings of ICCA2018, pages 304--311.

Win Pa Pa, Ye Kyaw Thu, Andrew M. Finch, and Eiichiro Sumita. 2016. A study of statistical machine translation methods for under resourced languages. In SLTU-2016, 5th Workshop on Spoken Language Technologies for Under-resourced languages, 9-12 May 2016, Yogyakarta, Indonesia, pages 250--257.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 311--318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Boonkwan Prachya and Supnithi Thepchai. 2013. Technical Report for The Network-based ASEAN Language Translation Public Service Project. Online Materials of Network-based ASEAN Languages Translation Public Service for Members, NECTEC.

Lucia Specia. 2011. Tutorial, Fundamental and New Approaches to Statistical Machine Translation. International Conference Recent Advances in Natural Language Processing.

Ye Kyaw Thu, Vichet Chea, Andrew M. Finch, Masao Utiyama, and Eiichiro Sumita. 2015. A large-scale study of statistical machine translation methods for khmer language. In Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation, PACLIC 29, Shanghai, China, October 30 - November 1, 2015.

Ye Kyaw Thu, Andrew Finch, Win Pa Pa, and Eiichiro Sumita. 2016. A large scale study of statistical machine translation methods for myanmar language. In Proceedings of SNLP2016.

Christoph Tillmann. 2004. A unigram orientation model for statistical machine translation. In Proceedings of HLT-NAACL 2004: Short Papers, pages 101--104, Boston, Massachusetts, USA. Association for Computational Linguistics.

String Similarity Measures for Burmese (Myanmar Language)

Khaing Hsu Wai
UTYCC, Myanmar
khainghsuwai@utycc.edu.mm

Ye Kyaw Thu
NECTEC, Thailand
ka2pluskha2@gmail.com

Hnin Aye Thant
UTYCC, Myanmar
hninayethant@utycc.edu.mm

Swe Zin Moe
UTYCC, Myanmar
swezinmoe.1011@gmail.com

Thepchai Supnithi
NECTEC, Thailand
thepchai.supnithi@nectec.or.th

Abstract

Measuring string similarity is useful for a broad range of applications. It plays an important role in machine learning, information retrieval, natural language processing, error encoding, and bioinformatics. Measuring string similarity is a fundamental operation of data science, important for data cleaning and integration. Real-world applications such as spell checking, duplicate finding, searching similar words, and retrieving tasks use string similarity. In this study, string similarity metrics have been calculated for Burmese (Myanmar language). The encoding table for Burmese has been built based on the pronunciation similarity of characters and vowel combination positions with a consonant. According to the table, strings and words are encoded. Similarity distance is measured between the dataset and query words. Previous string similarity approaches are not suitable for fuzzy string matching of tonal-based Burmese. Therefore, three mapping approaches are proposed in this study.

1 Introduction

Measuring string similarity is a fundamental operation in many applications of machine learning. It is widely studied in natural language processing (NLP). NLP applications such as text-to-speech, machine translation, spell checking, and information retrieval calculate string similarity metrics to find how similar the strings are. In other words, string similarity metrics help to find similar words according to a given query. Languages are interesting, and each language has its own features and writing systems. In the literature, several approaches have been proposed for string similarity. Most of them are character-based metrics and associated with English or European languages. For Burmese (language in Myanmar), we need to consider new approaches together with the existing string similarity metrics. Burmese is a tonal-based language and also a very rich language (Tun, 1990). It has 33 consonants, and the consonants are combined with vowels and medials to form syllables. In Burmese, not only one character can form a word (e.g., "က", "dance" in English) but also one syllable can form a word (e.g., "ကြိုက်", "like" in English). Additionally, there are many phonetically similar sounds of characters and words in Burmese. In our experiment, we proposed three mappings: phonetic mapping, sound mapping, and syllable combination mapping. We introduced a new approach based on the idea of Soundex, the best-known phonetic encoding algorithm, for retrieving phonetically similar words by calculating the string similarity distance. We have collected two datasets: one dataset contains the confusion pairs of words with real spelling mistakes, and another is a manually developed word similarity dataset. We evaluated six measures (cosine distance, Damerau–Levenshtein distance, Hamming distance, Jaccard distance, Jaro–Winkler distance, and Levenshtein distance) on two datasets, with and without the proposed three mappings. According to our results, all three mappings outperformed the existing approaches for retrieving Myanmar words with similar pronunciations.

2 Related Work

To the best of our knowledge, there is only one proposal that measured phonetic similarities of Myanmar Internationalized Domain Names (IDNs) (Ohnmar Htun, 2910). To retrieve

phonetically similar Myanmar IDNs, IPA (International Phonetic Alphabet)-Soundex functions were used for matching character values based on their phonetic similarities of Burmese. The normalized similarity method is capable of measuring similarity not only in a single language, but also in a cross-language comparison (Htun et al., 2011).

The Myanmar characters ultimately descend from a Brahmic script, either Kadamba or Pallava (Wikipedia, 2019). Likewise, most of the major Indian languages such as Devanagari (e.g., Hindi, Marathi, Nepali), Bengali (Bengali and Assamese), Gurmukhi (Punjabi), Gujarati, Oriya, Tamil, Telugu, Kannada, and Malayalam use scripts that are derived from the ancient Brahmi script. They have approximately the same arrangement of the alphabet, are highly phonetic in nature, and a computational phonetic model was proposed for them (Singh, 2006). It mainly consists of a model of phonology (including some orthographic features) based on a common alphabet of these scripts, numerical values assigned to these features, a stepped distance function (SDF), and an algorithm for aligning strings of feature vectors. The SDF is used to calculate the phonetic and orthographic similarity of two letters.

3 String Similarity Metrics

String similarity determines how similar two strings are. Various studies on string similarity has been carried out for different languages. In the literature, many methods to measure the similarity between strings have been proposed. Each method has its own features useful for NLP. Most similarity metrics are used to reduce minor typing or spelling errors in words or syllables in pronunciation. Based on the properties of operations, string similarity metrics can be divided into several groups.

Edit distance-based metrics estimate the number of operations needed to transform one string to another. A higher number of operations means less similarity between the two strings.

For token-based methods, the expected input is a set of tokens rather than complete strings. The purpose is to find similar tokens in both sets. A higher number of similar tokens means more similarity between the sets. A string can be transformed into a set of tokens by splitting it using a delimiter.

In sequence-based methods, the similarity is a factor of common substrings between the two strings. The algorithms try to find the longest sequence that is present in both strings. The more of these sequences found, the higher is the similarity score.

3.1 Levenshtein Distance

The Levenshtein distance (Levenshtein, 1966), also known as edit distance, returns the minimum number of edit operations in terms of the number of deletions, insertions, or substitutions required to transform the source string to the target string. A higher number of edit operations means less similarity between two strings. For example, the edit distance between "cat" and "dog" is 3. There are three edit operations needed to transform "cat" into "dog". For Myanmar language, "Fate"-"ကံ"(kan) and "ကန့်"(kan) (exact pronunciation with "ကံ" but different spelling and "kick, lake" in English), two edit operations are required. The Levenshtein distance is perfect for finding similarity of small strings, or for a small string and a big string, where the editing difference is expected to be a small number. The Levenshtein distance is defined recursively, as shown in Eq. (1).

$$dis_{a,b}(i,j) = \begin{cases} 0 & \text{if i=j=0} \\ i & \text{if j=0 and i>0} \\ j & \text{if i=0 and j>0} \\ min = \begin{cases} dis_{a,b}(i-1,j)+1 \\ dis_{a,b}(i,j-1)+1 & \text{otherwise} \\ dis_{a,b}(i-1,j-1)+1(a_i \neq a_j) \end{cases} \end{cases} \quad (1)$$

3.2 Damerau–Levenshtein Distance

The Damerau–Levenshtein distance is an algorithm that is similar to the Levenshtein distance; however, it additionally counts a transposition between adjacent characters as an edit operation (Damerau, 1964). For example, to transform string "CA" to string "ABC", the Levenshtein distance counts three edits, whereas the Damerau–Levenshtein distance is 2. For Burmese, the Levenshtein distance between "ကလေး"("baby") and "ခကလေး"(wrong spelling of "baby") is 3, whereas the Damerau-

Levenshtein distance is 2. Variations of this algorithm assign different weights to the edit based on the type of operation, phonetic similarities between the sounds typically represented by relevant characters, and other considerations.

3.3 Hamming Distance

The Hamming distance between two strings of equal length measures the number of positions with mismatching characters (Hamming, 1950). The Hamming distance only applies to strings of the same length. It is mostly used for error correction in fields such as telecommunication, cryptography, and coding theory. For example, the Hamming distance between "apple" and "grape" is 4, and the distance between "အဖေ"("father") and "အဖော"(wrong spelling of "father") is 1.

3.4 Jaro–Winkler Distance

The Jaro–Winkler distance is another string metric that measures an edit distance between two sequences (Jaro, 1989). The score ranges from 0 to 1, where 0 is "no similarity," and 1 is "exactly the same strings." The Jaro–Winkler distance is used to find duplicates in strings, because the only operation that it considers is to transpose the letters in a string. Eq. (2) describes the Jaro–Winkler distance d_j of two given strings s_1 and s_2, where m is the number of matching characters, and t is half of the number of transpositions.

$$d_j = \begin{cases} 0 & \text{if m=0} \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) & \text{otherwise} \end{cases} \quad (2)$$

3.5 Cosine Similarity

The cosine similarity between two vectors is a measure that calculates the cosine of the angle between them (Singhal, 2001). By calculating the cosine angle between the two vectors, we can decide if the vectors are pointing to the same direction or not. Two vectors with the same orientation have a cosine similarity of 1, which means that the two strings are equal. For two strings "ဇနီးမောင်နှံ"("husband and wife") and "ကလေး"("baby"), the cosine similarity is 0, but for "ဇနီးမောင်နှံ"("husband and wife") and "စနီးမောင်နှံ"(wrong spelling of "husband and wife"), the similarity distance is 0.75, which is nearly 1. Eq. (3) shows the formula of cosine similarity.

$$similarity(A,B) = \frac{A.B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}} \quad (3)$$

3.6 Jaccard Similarity

The Jaccard similarity measures similarities between sets (Jaccard, 1901). It is defined as the size of the intersection divided by the size of the union of two sets. For example, for sets $A = \{1, 2, 3\}$ and $B = \{1, 2, 4, 5\}$, the Jaccard similarity is 0.4. The Jaccard similarity is calculated according to the following equation.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cup B|} \quad (4)$$

3.7 Soundex Algorithm

The Soundex algorithm is a phonetic algorithm (Odell, 1956). It is based on how close two words are depending on their pronunciation. For example, the code for "Flower" and the code for "Flour" is 'F460" according to the Soundex encoding table, because they have the same pronunciation. Based on the idea of the Soundex algorithm, we propose three mappings for Burmese. All mappings aim to find words based on their phonetic similarity.

4 Proposed Mappings

String similarity algorithms have some difficulties with Burmese because it is a tonal-based language and is composed of vowels, consonants, and medials. With Myanmar alphabets, many words have the same pronunciation but different meanings (e.g., "ကံ", "luck" in English and "ကန်", "lake" in English). Moreover, some words have similar pronunciations and different meanings (e.g., "ခုနစ်", "seven" in English and "ခုနှစ်", "year" in English). To consider phonetically similar words, we propose three mapping tables for Myanmar words.

4.1 Phonetic Mapping

In our proposed methods, the first mapping is the phonetic mapping. Words with the same pronunciation are grouped together. For example, "ကလေး" and "ခလေး" have the same

pronunciation. Therefore, “က” (Ka) and “ခ” (Kha) are clustered to “က” (Ka) group. Likewise, other consonants with same pronunciation, such as “ဃ” (Ga) and “ဃ” (Gha), “ပ” (Pa) and “ဖ” (Pha), “ဗ” (Ba) and “ဘ” (Bha) are put together as groups, respectively, and some diacritics, such as “ွ” (Wa Hswe) and “ှ” (Ha Hto), tone marks such as “့” (Aukmyit), “်” (Myanmar sign Virama) are considered to be removed. Mapped characters are using both Myanmar and English alphabets for simple reading and an easier practical implementation. The details of the phonetic mapping table are shown in Table 1.

Char	Mapped Char	Char	Mapped Char
က ခ	က	ွ ှ	(delete)
ဂ ဃ	ဂ	ဣ ဤ ိ ီ ဥ	i
စ ဆ	စ	က် ပ် တ်	d
ဇ ဈ	ဇ	န် မ် ံ	n
ဋ တ	တ	ဲ ရ်	e
ဌ ထ	ထ	ဥ ဦ ု ူ	u
ဍ ဎ	ဍ	ာ ါ	r
ဏ နဏ	နဏ	ဧ ေ	a
ဒ ဓ	ဒ	့ း	(delete)
ပ ဖ	ပ	်	(delete)
ဗ ဘ	ဘ	ြ ျော် ြ ျော်	o
ယ ရ	ရ	၎င်း ၎	၎
လ ဠ	လ	၊ ။	s
သ ဿ	သ	ံ င် ဂ ဉ်	in
ျ ြ	y	?!.*-=#”<>[],+-	s

Table 1: Phonetic Mapping

4.2 Sound Mapping

The second mapping is the sound mapping. This mapping is similar to the phonetic mapping, but the main difference is in processing Myanmar consonants. As the name of the sound mapping suggests, consonants that have the same movements of mouth, lips, and tongue, are grouped. For example, “က ခ ဂ ဃ င ဟ အ” (Ka Kha Ga Gha Nga Ha A) are clustered to “က” (Ka) group, “ည ဉ” (NyaGyi NyaLay) are clustered to “ည” (Nya) group, “ပ ဖ ဘ ဗ မ” (Pa Pha Ba Bha Ma) are clustered to “ပ” (Pa) group, “ယ ရ” (YaPetLet YaGauk) are clustered to “ရ” (Ya) group. The details of the sound mapping are shown in Table 2.

Char	Mapped Char	Char	Mapped Char
က ခ ဂ ဃ င ဟ အ	က	ွ ှ	(delete)
ည ဉ	ည	့	(delete)
စ ဆ ဇ ဈ	စ	က် ပ် တ်	d
ဋ ဌ ဍ ဎ တ ထ ဒ ဓ န	တ	န် မ် ံ	n
ပ ဖ ဗ ဘ မ	ပ	ဲ ရ်	e
ယ ရ	ရ	ဥ ဦ ု ူ	u
လ ဠ	လ	ာ ါ	r
သ ဿ	သ	ဧ ေ	a
ျ ြ	y	့ း	(delete)
၊ ။	s	ြ ျော် ြ ျော်	o
၎င်း ၎	၎	ဣ ဤ ိ ီ ဥ	i
ံ င် ဂ ဉ်	in	?!.*-=#”<>[],+-	s

Table 2: Sound Mapping

4.3 Vowel Position Mapping

Myanmar writing system or word formation largely depends on the combination of left, right, upper, and lower characters to a consonant (i.e., consonant clusters or syllable). Here, left, right, upper, and lower characters mean dependent vowels, directives, and subscript consonants that are always written with a consonant (Thu and Urano, 2007) according to their written positions.

The third proposed mapping is based on the syllable formation in Burmese, we call it the vowel position mapping. Thus, the vowels written on the left side of the consonant are under the left (l) group, the right-side vowels are under the (r) group, the upper vowels are under the (u) group, the lower vowels are under the (d) group. If we represent the core concept of the vowel position mapping with Python programming, the code for building a dictionary variable named “map3_dict” will be as follows:

```
map3_dict = [
('[က-အ]', 'c'),
('ျ |ြ', 'y'),
('ေ', 'l'),
('ိ |ီ |ဲ |ံ ', 'u'),
('ွ |ှ |ု |ူ , 'd'),
('ာ |ါ |့ |း ', 'r'),
]
```

Here, “c” is used for consonants, “y” for medial characters “ျ” and “ြ”, “l” for the “left”, “u” for “upper”, “d” for “down” or “lower”, and “r” for “right”-side characters. The details of the vowel position mapping are shown in Table 3. This mapping is designed for re-

trieving Myanmar words that have a similar vowel combination structure.

Char	Mapped Char	Char	Mapped Char
a-z A-Z	F	က-အ	c
ျ ြ	y	့	p
ေ	l	ာ ါ ့ း	r
ိ ီ ဲ ံ	u	် ္ ် ူ	d
်	k	၊ ။	s
ဣ ဤ ဥ ဦ ဧ ဩ ဪ ဿ ၌ ၍ ၏	i	?!.*-=#"<>[],+-	$
၀-၉	n	0-9	D

Table 3: Vowel Position Mapping

5 Experiments

We compare 6 similarity measures on our three mappings. They are Levenshtein, Hamming, Jaro–Winkler, Damerau–Levenshtein, cosine, and Jaccard similarities. We conduct two experiments with two datasets that we have collected.

5.1 Datasets

We have collected two datasets: *Spelling Mistake Confusion Pairs* and *Word Similarity Dataset*.

5.1.1 Spelling Mistake Confusion Pairs

The dataset of spelling mistake confusion pairs was developed based on real-world spelling errors. Mainly, we collected general-domain text, especially from Myanmar news and social media websites, such as BBC (British Broadcasting Corporation) Myanmar, VOA (Voice of America) Myanmar, Facebook, and emails during March 2018 and July 2019. The dataset contains 2,381 pairs (i.e., 4762 words). Some examples of confusion pairs are as follows:

- ကိုကိုကြီး - ကိုကိုကြီး
- ကောင်းကောငး် - ကောင်းကောင်း
- ကောင်းကျပါတယ် - ကောင်းကြပါတယ်
- ခွင့်မလွတ်ပါနဲ့ - ခွင့်မလွှတ်ပါနဲ့
- ငါ့စီ - ငါ့ဆီ
- စီးပွားေရ့း - စီးပွားရေး
- စွဲချက်တင်နိုင်သောကြောင့်-စွဲချက်တင်နိုင်သောကြောင့်
- တောင်ပန်အပ်ပါတယ် - တောင်းပန်အပ်ပါတယ်
- တိုင်ပြည်ချစ်စိတ် - တိုင်းပြည်ချစ်စိတ်
- ဒေါ်အောင်ဆနး်စုကြည်-ဒေါ်အောင်ဆန်းစုကြည်
- နက်နက်ရှိုင်းရှိုင်း - နက်နက်ရှိုင်းရှိုင်း
- ပြဿနာတက်မှာဆိုးပြီး - ပြဿနာတက်မှာစိုးပြီး
- ၂၀၁၀ - ၂၀၂၀
- ဝူးရှူး - ဝူရှူး
- အဆောက်အဦ - အဆောက်အဦး

During the dataset collection, we found that some of the spelling mistakes are caused by encoding conversion between partial Unicode named "Zawgyi" and other Unicode fonts such as "Myanmar3" and "Padauk" (e.g., ကိုကိုကြီး - ကိုကိုကြီး, တနလာဂနေ့ - တနလ်ာနေ့, နိုင်ငံရေး၏် - နိုင်ငံရေး၏). Moreover, the spelling mistakes based on pronunciation similarity (e.g., ကျေးပွန်းစွား - ကျေးပွန်စွာ, ငါ့စီ - ငါ့ဆီ, ပြဿနာတက်မှာဆိုးပြီး - ပြဿနာတက်မှာစိုးပြီး) and shape similarity (i.e., glyph) of Myanmar characters are also found (e.g., စီးပွားေရ့း - စီးပွားရေး, ဝူးရှူး - ဝူရှူး, အဆောက်အဦ - အဆောက်အဦး). All the confusion pairs generally have one-to-one relationship between misspelled and correct words; thus, we assumed it is very useful for evaluating on our three mappings. However, this dataset has few homophones and rhyme words; therefore, it is not suitable for measuring pronunciation similarity.

5.1.2 Similar Pronunciation Dataset

We developed the similar pronunciation dataset to evaluate similarity scores provided by our three mappings. Based on the correct Myanmar word, we manually added one homophone and three more rhyme words, such as "Hat:Bat", "Fun:Sun", "Honey:Money". For example, the first column word မြူးတူး("festivity" in English) is the correct word, the second column မြူးထူး is the homophone word, and the other following columns ဂျူးဖူး, ကူးလူး, and ပြူးတူး are three rhyme words of the first column word (see Table 4). We collected 200 pairs for the similar pronunciation dataset, with 1,000 words in total.

Correct Word	Homophone	Rhyme1	Rhyme2	Rhyme3
မြူးတူး	မျူးထူး	ဂျူးဖူး	ကူးလူး	ပြူးတူး
ပြဿနာ	ပြဿနာ	ရှုစမ်း	ကြာပန်း	ကျင်းနီး
တချို့	တစ်ချို့	အချို့	သချို့	နှစ်ချို့
ကြွေးမြီ	ကျွေးမြီ	ခွေးမြီး	ကြေးမုံ	ချွေးသီး
ဂဃနဏာ	ဂဂနန	ခခယယ	မမထထ	ခခရရ
လက်ရွေးစင်	လက်ရွေးစဉ်	လက်ယွေးစင်	ရက်ရွေးစင်	လက်ရွေးဇင်

Table 4: Examples from the Similar Pronunciation Dataset

Examples for how our three proposed mappings work can be seen as the following table.

Phonetic Mapping	Sound Mapping	Vowel Position Mapping
ပစ္စည်း –> ပစၹi	ပစ္စည်း–> ပစၹi	ပစ္စည်း –> ccpcckr
ပစ်စည်း –> ပစၹi	ပစ်စည်း –> ပစၹi	ပစ်စည်း –> cckcckr

Table 5: Examples of Three Proposed Mappings

5.2 Evaluation

For the evaluation, we measured string similarity on each pair from both original datasets: "Spelling Mistake Confusion Pairs" and "Similar Pronunciation Dataset". Next, we encoded or converted the original data into our 3 mappings and measured string similarity again. Finally, we counted the correct words or similar words based on the three thresholds "<=1", "<=2", and "<=3" for "Levenshtein, Damerau–Levenshtein, and Hamming distance measures" and ">=0.9", ">=0.7", and ">=0.5" for "Jaro–Winkler, cosine, and Jaccard distance measures".

6 Results and Discussion

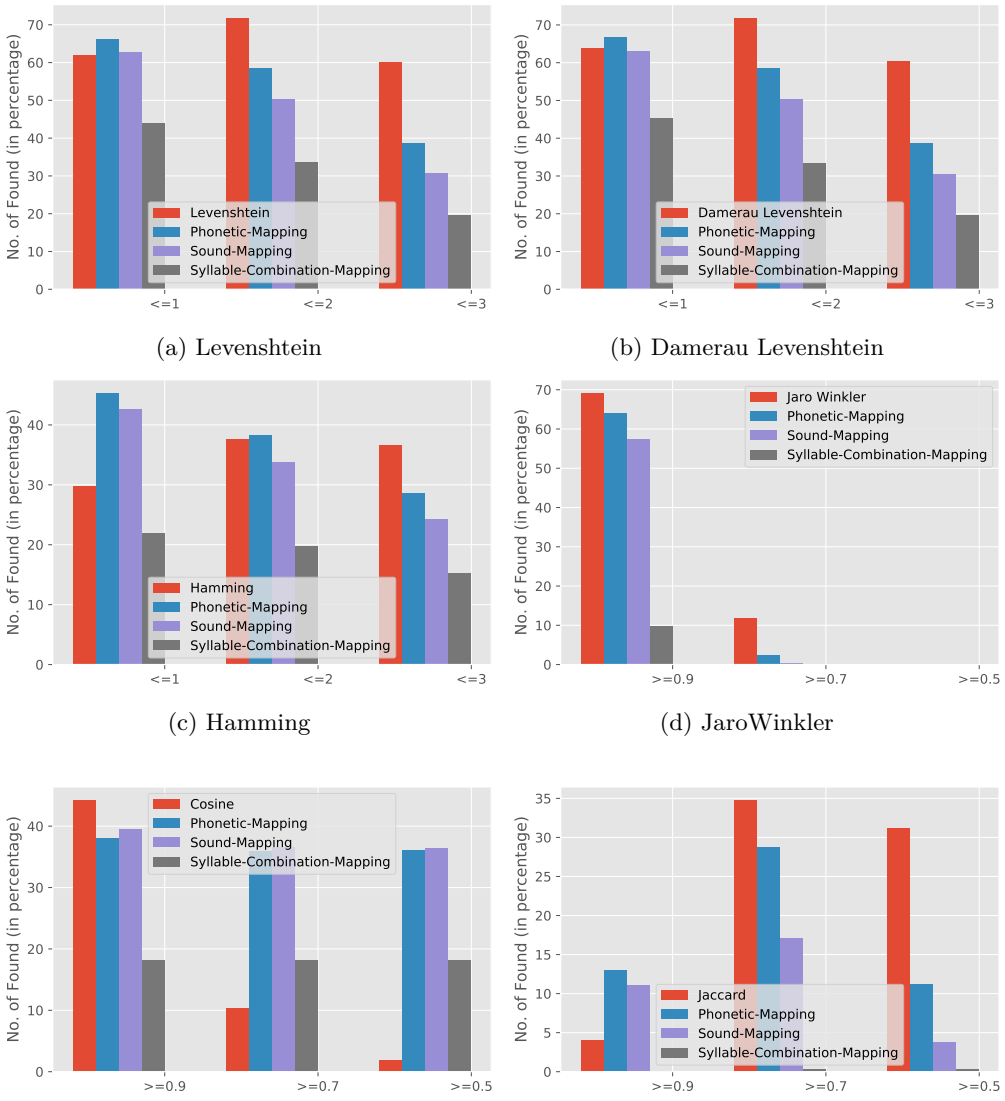
The number of correct words found for six similarity measures on the "Spelling Mistake Confusion Pairs dataset" is shown in Figure 1. According to these experimental results, our phonetic mapping gave a better word correction rate than four existing distance measures (Levenshtein, Damerau–Levenshtein, Hamming, and Jaccard) for threshold <=1 or >=0.9. Similarly, the sound mapping also achieved higher or comparable results, except for the Jaro–Winkler and cosine similarity. On the other hand, the vowel position mapping approach obtained the lowest correction rate for all thresholds.

For thresholds "<=2" and "<=3" (">=0.7", ">=0.5" for Jaro–Winkler and cosine similarity), generally, all proposed mappings are lower than raw Myanmar text input. However, we found that the phonetic mapping and sound mapping matched more correct words from the "Spelling Mistake Confusion Pairs" dataset for Hamming and cosine similarities.

According to these experimental results, our new two mappings (phonetic and sound mappings) are applicable for string similarity measurement on spelling mistake confusion words. Moreover, based on the current results for thresholds "<=2" and "<=3" (or ">=0.7" and ">=0.5"), we clearly found that the vowel position mapping is able to retrieve approximately 50% of the correct words for Levenshtein, Damerau–Levenshtein, Hamming, and cosine similarities.

The results of retrieving similar pronunciation words, such as homophones and rhyme words, with six similarity measures on the "Similar Pronunciation Dataset" is shown in Figure 2. As we expected, two of our proposed mappings, phonetic mapping and sound mapping, achieved the highest number of found errors for all thresholds of Levenshtein, Damerau–Levenshtein, Hamming, Jaro–Winkler, cosine, and Jaccard similarities. Additionally, the vowel position mapping also obtained the highest or comparable results for existing five distance measures, except for the Jaccard distance measure.

We did a detailed analysis on distance values, and we found that our proposed three mappings have a zero distance value (i.e., no distance value) for some similarly pronounced words. For example, the string similarity distances for the word လက်ရွေးစင် and similar pronunciation and rhyme words လက်ရွေးစဉ်, လက်ယွေးစင်, ရက်ရွေးစင် and လက်ရွေးဇင် for Levenshtein and our three mappings for the threshold "<=1" are shown in Table 6. Moreover, our three mappings retrieved similar words well, compared with inputting raw Myanmar text. For example, although Levenshtein distance (for the threshold "<=1") retrieved only one similar word of လွင့်စဉ် ("scatter" in English), our three mappings were able to retrieve three more similar words လွှင့်စင်, လွှင့်ဇင် and လွန့်စင် (see Table 7). One more example of cosine and our three mappings' string



(a) Levenshtein (b) Damerau Levenshtein

(c) Hamming (d) JaroWinkler

(e) Cosine (f) Jaccard

Figure 1: Results with the spelling-mistake confusion dataset

similarity distances of the word အကဲခတ် (“to assess” in English) (for threshold “>=0.9”) can be seen in Table 8. Here, “N\A” means “Not Applicable”, and the expression is not contained in the threshold distance.

Word - Similar Word	Levenshtein	Pronunciation	Sound	Vowel
လက်ရွေးစင် လက်ရွေးစဉ်	1	0	1	0
လက်ရွေးစင် လက်ယွေးစင်	1	0	0	0
လက်ရွေးစင် ရက်ရွေးစင်	1	1	1	0
လက်ရွေးစင် လက်ရွေးဇင်	1	1	0	0

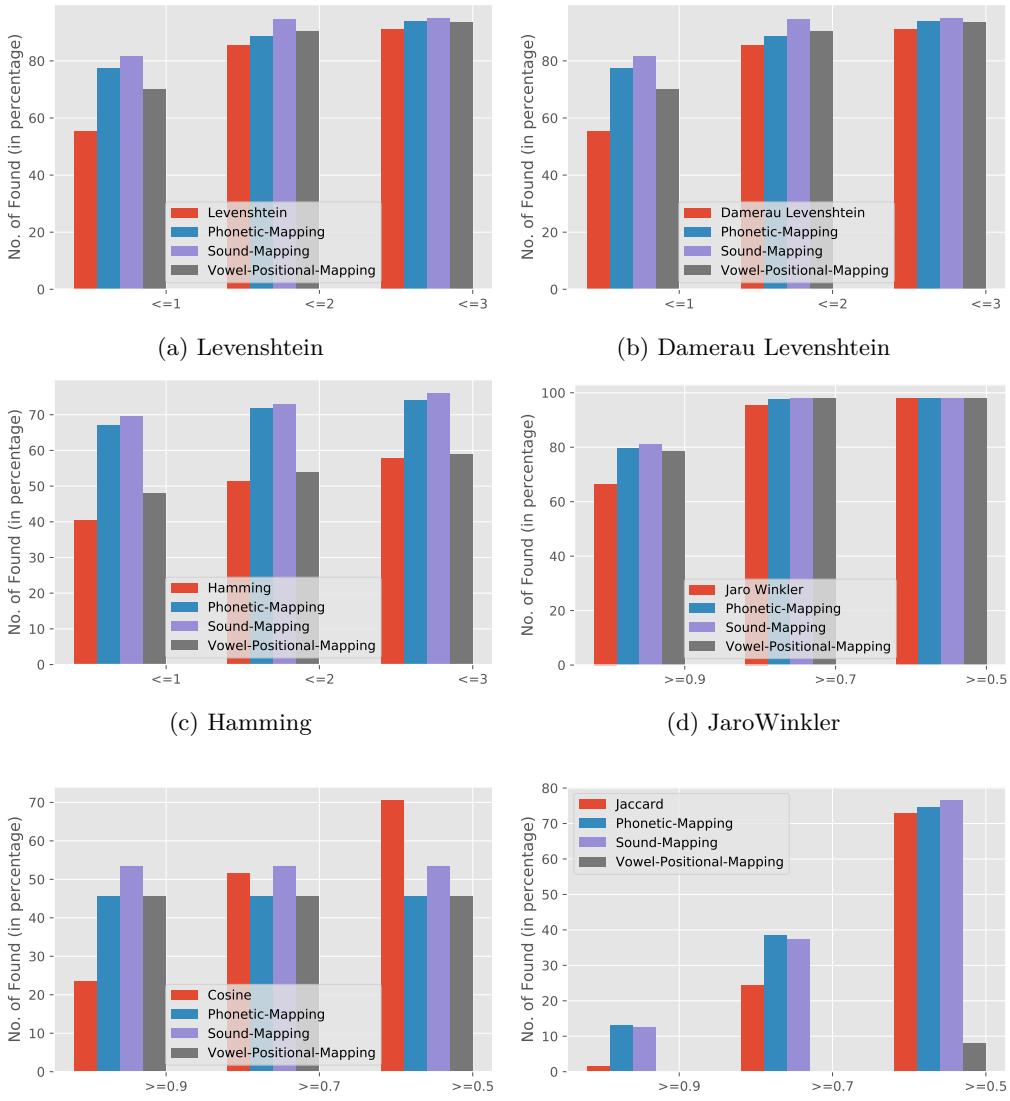
Table 6: String similarity distances for the word “လက်ရွေးစင်” (“selection”) in English

Word - Similar Word	Levenshtein	Pronunciation	Sound	Vowel
လွှင့်စဉ် လွှင့်စင်	1	0	1	0
လွှင့်စဉ် လျှင့်စင်	N/A	0	1	1
လွှင့်စဉ် လွှင့်ဇင်	N/A	1	1	0
လွှင့်စဉ် လွန့်စင်	N/A	1	1	0

Table 7: String similarity distances for the word “လွှင့်စဉ်” (“scatter” in English)

7 Conclusion

In this paper, we have presented the first study of the string similarity measurement based on the pronunciation similarities for Burmese. We proposed three new mappings (phonetic mapping, sound mapping, and vowel position Mapping) and proved a better re-



(a) Levenshtein

(b) Damerau Levenshtein

(c) Hamming

(d) JaroWinkler

(e) Cosine

(f) Jaccard

Figure 2: Results with the similar pronunciation dataset

Word - Similar Word	Cosine	Pronunciation	Sound	Vowel
အကဲခတ် အကဲခပ်	N/A	1.0	1.0	1.0
အကဲခတ် အကဲဆတ်	N/A	N/A	N/A	1.0
အကဲခတ် အမြဲတက်	N/A	N/A	N/A	N/A
အကဲခတ် မဆဲတတ်	N/A	N/A	N/A	1.0

Table 8: String similarity distances for the word "အကဲခတ်" ("to assess" in English)

trieving of similarly pronounced words, homophones, and rhyme words. Moreover, the phonetic mapping and sound mapping are applicable for spelling correction by string similarity measurement of Burmese under the threshold "<=1". In the future work, we plan to expand the two datasets and conduct string similarity experiments to confirm our current mapping tables.

References

Fred J. Damerau. 1964. A technique for computer detection and correction of spelling errors. *Commun. ACM*, 7(3):171–176.

R. W. Hamming. 1950. Error Detecting and Error Correcting Codes. *Bell System Techincal Journal*, 29:147–160.

Ohnmar Htun, Shigeaki Kodama, and Yoshiki Mikami. 2011. Cross-language phonetic similarity measure on terms appeared in asian languages. *International Journal of Intelligent Information Processing*, Volume 2:9 21.

Paul Jaccard. 1901. Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines. *Bulletin de la Societe Vaudoise des Sciences Naturelles*, 37:241–72.

Matthew A. Jaro. 1989. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, 84(406):414–420.

VI Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, Vol. 10:707.

Margaret King Odell. 1956. The profit in records management systems. page 20: 20, New York.

Yoshiki Mikami Ohnmar Htun, Shigeki Kodama. 2910. Measuring phonetic similarities in myanmar idns. March 2010, pages Page–129–135.

Anil Kumar Singh. 2006. A computational phonetic model for indian language scripts.

Amit Singhal. 2001. Modern information retrieval: A brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 24(4):35–43.

Ye Kyaw Thu and Yoshiyori Urano. 2007. Positional mapping: keyboard mapping based on characters writing positions for mobile devices. In *Proceedings of the 9th International Conference on Multimodal Interfaces, ICMI 2007, Nagoya, Aichi, Japan, November 12-15, 2007*, pages 110–117.

Thein Tun. 1990. The domain of tones in burmese. pages pp. 406–411.

Wikipedia. 2019. Burmese language wikipedia page. [Online; accessed 18-July-2019].

An Inferential Phonological Connectionist Approach to the Perception of English-Assimilated Speech

Hiba Zaidi
Department of Foreign Languages
Yahia-Fares University
Medea, Algeria
hibato92@hotmail.com

Abstract

This research is an attempt to investigate the perceptual system processing of English connected speech. More specifically, it attempts to offer an experimental study to show the impact of the phonological context and specific English learning experience on non-native listeners perception of a set of assimilated-English speech forms inserted in pair of words and sentential contexts. Given that assimilation of place as a regular phonological process shaping English casual speech, seems to pose a notorious problem for the perceptual system in general and for non-native listeners to English continuous speech in particular. This triggers the current research investigation of a networks performance on English assimilated forms so as to have rather measurable and testable results.

1 Introduction

Connectionism conventions of human cognitive phenomena, according to (Rogers, 2009), appeal for a propagated activation through interconnected units of the system. In this realm, speech processing area (speech perception) has caught a large area of computational connectionist models interest. Taking that speech perception is organized around a lexical access process (Traxler, 2012), (JEFFREY and JAMES, 1988), thus, lexical mismatch intolerance is highly invoked while phonological variations in English speech production are widely attested. More pointedly, assimilation of place is regarded as a regular process shaping English connected speech, and is disruptive to the recognition system function .i.e., it interrupts the lexical access process through which speech inputs are matched to their lexical representational units stored in the long-term memory. Consequently, a misperception and sometimes lexical ambiguity of the speech arise, as argued by (Mec-Queen et al., 1999). It incurs a mismatch between the incoming variant segments and their lexical representational entries. This prompts the question as how the system handles such perceptual perturbation Inferential Phonological Connectionist Model (Gaskell and Gareth, 2003) (Mg and Wd., 1996a) (Mg and Wd., 1996b) (Marslen, 1998) seems to be central for addressing place-assimilated speech drawing on two main notions. First, the perceptual system processes the assimilated forms through a gradual and regressive mapping of surface (output unit) onto underlying (input unit) by making use of SNR architecture (Mg and Wd., 1996a). Second, the system is very sensitive to the phonological conditions surrounding the assimilated forms and tends to infer the phonological rules regulating such variations. However, driving on the expectation that the network might have an active role in shaping the phonological rules regulating the assimilation process rather than simply inferring them, this incites the current research to approach the issue concerned based on a modified SNR version . In light of the aforementioned predictions, this research will proffer an experimental study on a network performance on a set of assimilated forms embedded in pair words and sentential contexts. The research will draw from the experimental design to guide the different empirical phases, including the preliminary pre-test, treatment and post-test phases.

2 Phonological Inferential connectionist Model

Phonological Inferential Model (Gaskell and Gareth, 2003) (Mg and Wd., 1996a) (Mg and Wd., 1996b) (Marslen, 1998) is a computational approach directed to speech processing area. It has been developed to find conceivable answers to recurrent questions faced in cognitive theories of speech perception, as how speech is pro-

cessed. The model evaluates the interconnection between a set of phonological rules and given phonological contexts of a speech stream, in the sense that, as (Mg and Wd., 1996b) pointed, these rules are context-sensitive as they specify the context they can apply to (p.288). For example, the likely occurrence of / k/ as a variant of the lexical /t/ in the word thatking to be realized as /kkI/ at the surface level, is permissible only if the following segment has a velar place feature like /k/, providing a source for a velar place feature to be repressively (leftward) assimilated. Given the crucial property of phonological rules as being phonological context-sensitive rules, this amount to characterize the model for a heuristic account for speech perception for the reason that it calls in simultaneously both listeners specific phonological experience (acquired phonological rules) alongside the range of phonological context surrounded-variation. As far as speech processing area is concerned, the current model (IPM) holds for three main assumptions: first, a successful completion of the lexical access process is tied with the perceptual system gradual learning to recognize speech, in the sense that, as argued by (Mg and Wd., 1996b), that the process of learning to compensate for phonological variant forms (assimilated forms) depends on the gradual connection of the constraints involved in the goodness-of-fit between the variable output and input computation in lexical access(p.416). Second, they contend that unassimilated forms are more likely to be recognized by the recognition system as being the underlying forms and, thereby, inferring the assimilated forms (p.416). This is motivated by the fact that the phonological rules regulating assimilation of place are constrained by the phonological context viability for assimilation. For instance, the following phonological rules regulating six assimilatory patterns are central to the second assumption.

Based on the aforementioned IPM assumption, simplified network recognition (SNR) has been developed as figure (2) shows:

The SNR architecture has been generated to examine the lexical access process using a simple recurrent network trained on mapping a speech phoneme input, undergoing assimilation variation onto an output window, exploring a back propagation-algorithm. However, this research, as differently from Gaskells SNR hypothetical and

t → p / -# (p, b, m)
d → b / -# (p, b, m)
n → m / -# (p, b, m)
t → k / -# (k, g)
d → g / -# (k, g)
n → ŋ / -# (k, g)

Figure 1: Six Assimilatory Patterns (Gaskell et al., 1996)

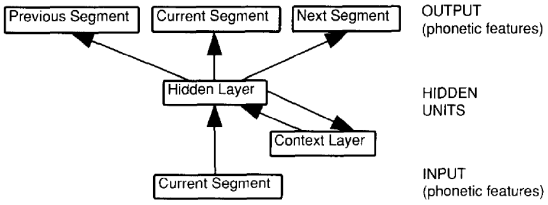


Figure 2: SNR Architecture (Gaskell et al., 1996)

practical orientations, tends to account for the assimilation processing by altering the current architecture drawing on the following perspective: the perceptual speech might have an active role in shaping regular and systematic phonological changes (e.g., assimilation of place process) and, therefore, would be able to shape the phonological rules (represented in figure 1) governing assimilation of place. Yet, the viability of the current perspective seems to be bound with the specific learning experience that the system undertakes; the system should be trained on mapping from underlying segments onto their surface assimilated segments. By doing so, the system is taken to learn the conditions in which the different six patterns of assimilations take place and, therefore, learn to recognize the respective phonological rules governing each pattern. This assumption is central to the current research revised SNR structure:

The proposed modified structure incites the research procedure described in the next section.

3 Research Procedure

The network will be trained on mapping from underlying unassimilated segments onto their surface assimilated variants in the prediction that the network would recognize the phonological rules underlying the assimilation of place pattern for the six cases as represented in figure (1). Moreover, following (Mg and Wd., 1996b) practice, in the purpose of stimulating the network to learn

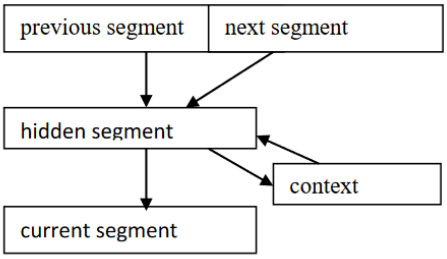


Figure 3: SNR Modified Structure

the associations between unassimilated segments (underlying segments) and their neighboring segments, we will incorporate a large corpus, Brown Corpus (H and N, 1960), whereby place assimilation will be artificially introduced into the corpus. A random selection of 50 % of coronal segments (/t/, /d/, /n/) contained in the data found in phonological contexts triggering assimilation (/p/, /b/, /m/, /k/, /g/) will be carried, then changing their place features so that to match them with the following phonological rules listed in figure (1). Moreover, the network will be trained on phonetic transcription of the artificially introduced assimilated words, as most of transcriptions of speech do not show phonetic representations of phonologically (assimilated) words. For this purpose, we will employ the LUND corpus of Stvartvik and Quirk (1980). We will convert the orthographic forms into phonetic forms employing a translation program (phonetizer program), and if any errors occur, they will be manually corrected. The network will be trained on 50 sweeps among the whole corpus, with the identity of the assimilated segments altering between the sweeps, employing TLearn program software.

4 Research Design

In the aim of examining the variant speech perception in terms of the phonological related-experience aspect, the process through which the perceptual system applies phonological rules to a range of underlying speech forms so that to recognize the surface forms, this research will assign an identification-based test to the network. Our predictions on the network performance pertain to the thought that it would make use of the recurrent links to apply the set of phonological rules regulating place assimilation to the data. For instance, the network is expected to shape the rule that for an underlying coronal (/t/) to be surfaced as (/p/), it must be followed by a segment with a bilabial non-coronal feature (/p /, /b/ or /m/). To test the network perceptual performance, we will use a set of two-word stimuli for 15 tokens and other 15 sentence-fillers to the trained network segment by segment, whereby the first word and second word final segments are deleted, however, the following words initial segments alter between triggering regressive assimilation of place to take place or blocking the process. For instance, the two-word stimuli /kwaipein / and /kwaisein/ derived from the word quite pain have their final coronal /t/ deleted ,however the /p /segment in the following word triggers the assimilation of the deleted segment as it provides a phonological context viable for the process to take place, while the second context, where /s/ is the initial consonant in the following word /sein/ blocks the assimilation of the deleted segment regarding the phonological context unviability for the process to occur. Each time, when the two-word stimuli for the 15 tokens are presented, they will be synchronized with the prime display (the tokens underlying forms). The network performance on the final segment in the first and second word will be measured at two points: one when the final deleted segment is recovered, whereby the output of the current segment window is recorded and examined, the other one will be measured when the following segments, presented along with the prime and the activations of the previous segment window were recorded so that to enable examination of the network identification of the final segment in the first word. Moreover, in the aim of testing the phonological context effect on the network perceptual performance, we will assign a discrimination-based test to the network with the use of other 15 carrier pair of words. However, this time the tokens will be presented with three modification types of the following phonological contexts: segments presented in their assimilated forms and, therefore, inserted in a minimally unviable phonological context (/kwaipein/), segments presented in maximally unviable phonological conditions, whereby the final segment in the first word is unassimilated and the initial segment in the following word generates non-lexical input (/kwaitkein/), and segments inserted in a viable phonological context for assimilation whereby the final segment is presented as unchanged and the initial segment in the neighboring word triggers

the assimilation process (/kwaitpein/). Each type of these three variations will be presented twice to the network, followed by a mean coronal score across segments featured with velar or labial resultant place, to be used later in the analysis.

5 Conclusion

Given that assimilated-speech variants (speech sound-streams ending in the coronal/ t /, /d /, /n/ are neutralize respectively into /p/, b/,/m/,/k/ ,/g/ ,//, when followed by one of the non-coronals /p/, /b/, /m/ , /k/, /g/), this invokes a propagated activation throughout the interconnected input, output and hidden units of the system. In parallel with this, and in light of the SNR structure, which explores a back propagation-algorithm, the system tends to infer, through a gradual training on mapping from the surface (output) to the underlying (input) unit, the phonological rules underlying the assimilation process. Gaskell (Mg and Wd., 1996b) maintained, this simple architecture allows the network to learn generalizations based on the statistics of the speech stream, using these generalizations to improve the networks performance in the prediction of upcoming phoneme (p.415). However, the current research tends to improve the networks performance by resorting to a different mapping; training the network to map from input to output units, henceforth, optimizing the system to shape and accommodate the range of phonological variations to the process of the lexical access regularities.

Acknowledgments

The author would like to thank the almighty God for having a number of issues addressed in this paper.

References

Gaskell and Gareth. 2003. Modeling regressive and progressive effects of assimilation in speech perception. *Journal of Phonetics*, 31 (3):447–463.

Kucera H and Francis N. 1960. Brown corpus. *US Cooperative Research Program Office of Education and Brown Corpus*, (p41-50).

ELMAN JEFFREY and McClelland JAMES. 1988. Cognitive penetration of the mechanisms of perception: Compensation for coarticulation of lexically restored phonemes. *Journal of Memory and Language*, 2.

Wilson WD Marslen. 1998. Mechanism of phonological inference in speech perception. *Journal of Experimental Psychology: Human Perception and Performance*, 24(2).

James A MecQueen, Norris D, and Cutler A. 1999. Lexical influence in phonetic decision mismatches making: Evidence from subcategorical mismatches. *Journal of Experimental Psychology: Human Perception and Performance*, 25(2).

Gaskell Mg and Marslen-Wilson Wd. 1996a. Phonological variation and inference in lexical access. *Journal of Experimental Psychology: Human Perception and Performance*, 22(1):144–158.

Gaskell Mg and Marslen-Wilson Wd. 1996b. Phonological variation in speech perception: A connectionist approach. York, England.

Timothy T Rogers. 2009. Connectionist models. *Encyclopedia of Neuroscience*, (p75-82).

Matthew J Traxler. 2012. *Introduction to Psycholinguistic Understanding of Language Science*. Wiley-Black Well, India.

Improving NER Models by exploiting Named Entity Gazetteer as External Knowledge

Atefeh Zafarian, Habibollah Asghari
ICT Research Institute
Academic Center for Education,
Culture and Research (ACECR)
Tehran, Iran
{Zafarian, habib.asghari }@ictrc.ac.ir

Abstract

This paper proposes a supervised NER model based on gazetteer for NSURL-2019 Task 7: Named Entity Recognition (NER) in Farsi. Supervised methods generate acceptable results in many Natural Language Processing tasks such as Named Entity Recognition (NER). Since these methods are domain-based, so their quality is related to the volume and the domain of training data. External knowledge can help the supervised methods to compensate this deficiency. In this paper, we use an unlabeled corpus as external knowledge to extract a named entity gazetteer for improving the performance of NER systems. We apply a supervised NER model on the unlabeled corpus to extract named entities with high probability. Finally we train a new NER model by using the gazetteer as a new feature to be employed with other features. The results show that the performance of NER model exploiting the gazetteer outperforms the ordinary models.

1 Introduction

This paper proposes a NER model for (Taghizadeh et al., 2019). NER systems extract important names from text such as person, location and organization. Some NER systems may cover other tags such as time, date, money etc. due to the type of the information that we expect to extract from the text.

Many of the previous works use supervised methods for constructing high performance NER models. They generate good results but only on their specific domain, but if the domain changes, they won't work efficiently. For compensation, some of the researches used external knowledge such as entity dictionaries or gazetteers. Gazetteers contain named entities that researchers add them as an external knowledge for improving the performance of NER model. However, generating and maintaining high-quality gazetteers is very time consuming. There are some methods that have been proposed for solving this problem by automatically extracting gazetteers from external knowledge for example Torisawa (2007). In a research investigated by Torisawa, (2007), they have extracted NEs from Wikipedia by automatic methods. Although extracted information of Wikipedia as a gazetteer is useful for training NER models, they don't cover all of the new entities because of rapid changes in the information content. Moreover, they cannot extract all of the tags and only focus on a limited set of tags such as person, location and organization names. However, many of the applications need more tags.

In this paper, we propose an automatic method for generating a named entity gazetteer from a big unlabeled corpus. At first, we use a supervised NER model to decode unlabeled corpus. At the second step, we extract a high-confidence named entity list from the unlabeled corpus as an entity gazetteer. Finally we add the gazetteer as a new feature to the model and retrain our NER model to generate a new one.

For generating the corpus for both the NER model and the gazetteer, we use the news data from Persian news agencies. This approach is beneficial and improves the performance of NER model because it adds the newest information from recently released news to our model. Moreover, the gazetteer is designed in such a way to be in similar domain with the NER corpus. So, it gives a better performance in comparison to Wikipedia resource because it contains most recent information from the news text.

The experiments in this paper are conducted in Persian and the data set is a large NER corpus, coming from the NSURL shared task. We show our results in phrase level and word level for a 3 classes and a 7 classes NER system. We also show the achieved results for all of the tags. Our results show an acceptable accuracy in F-score and a good result in precision. Also we got a good accuracy in new tags such as 'Time', 'Date', 'Money' and 'Percent'.

The paper is organized as follow: in the second section, an overview of the pervious works in exploiting gazetteers to enhance the performance of NER models will be presented. Section 3 comes with our proposed method. In section 4, the experimental setup is explained including the data set and evaluation measures. In section 5, our experiments and results are thoroughly described. Conclusion and future works are described in the last section.

2 Related Work

There are different approaches for generating NER models. Some of them use external knowledge as a feature to improve their model. For example, Torisawa (2007) retrieves the corresponding Wikipedia entry for each candidate word sequence and improves the NER system by the candidates. (Nothman et al., 2008) transforms the Wikipedia link into Named Entity Recognition by classifying the target Wikipedia pages into common entity types. Cucerzan (2007) have employed Wikipedia in order to support a Named Entity Recognition and disambiguate extracted named entities. (Bøhn and Nørvag, 2010) have applied Wikipedia contents to automatically generate an entity dictionary to connect the same named entity to the same tag. In a research investigated by (Nadeau et al., 2006) they proposed an unsupervised named entity Recognition by automatically extracting gazetteers from a large amounts of text. (Toral and Monachini, 2008) improved the performance of a named entity recognition by using external knowledge. (Etzioni et al., 2005) focused on automatic extraction from the Web for improving a Named Entity Recognition system. It should be noted that some researches have shown that larger NE lists do not necessarily correspond to increased NER performance (Mikheev et al., 1999).

3 Proposed Method

Our method includes five steps as follow:

- Preprocessing of the text.
- Training a CRF-based NER model.
- Crawling a large amount of news from Persian news agencies for generating an unlabeled corpus.
- Applying NER model on the unlabeled corpus and extracting high-confidence named entities as a gazetteer.
- Adding the gazetteer to CRF-based model and training the new model.
- In the following subsections we will describe the above mentioned steps in detail.

3.1 Preprocess

At the first step, we preprocess the NSURL corpus. We use Parsivar tools for text preprocessing (Mohtaj et al., 2018). There are some problems in the corpus; for example the whole of some sentences in the corpus were tagged as a single named entity. We remove the sentences because it increases the runtime and has negative effect on the results. Furthermore, we apply a normalizer on the corpus to unify the character codes.

3.2 Training the NER Model

We use CRF algorithm for training the model. Because of the supervised algorithm we used, it gives a high performance model. The tool that has been employed is CRF-based Stanford Named Entity Tagger. It presents good facilities for define NER features.

We checked different features for NER model and identified a series of n-gram features such as the assigned class of the word, the word itself and the previous and next words as best features for training the model. Table 1 shows the feature set used for our proposed model.

Description	Feature
Current Word	W3
Left Word	W2
Right Word	W4
Left Tag	T2
Two Left Words	W1W2
Two Left Tags	T1T2

Table 1: Feature Set.

3.3 Generate Unlabeled Data

As mentioned before, since the domain of NSURL corpus is from Persian news, so we use the text from Persian news for making unlabeled corpus. We crawl some popular news agencies and extract news from different categories. We focus on the domain of training corpus; for example if the training corpus contains only the text in sport domain, we crawl only sports news. Then, we apply a preprocessing tool on unlabeled Corpus and tokenize and normalize the sentences and remove very short and very long sentences.

3.4 Generate Named Entity Gazetteer

For generating named entity gazetteer, we decode unlabeled corpus with our NER model and extract words with high probability. For extracting these entities, we also consider sentence confidence using following the equation (Zafarian et al., 2015).

$$sentence\ confidence = \frac{\sum_{word \in i} word\ confidence}{\max(10, sentenc\ length)}$$

If the word confidence and sentence confidence are both reliable, we extract entities from that sentence.

3.5 Retrain NER Model

Finally, we add the named entity gazetteer as a new feature to our proposed NER model and re-train the model with this new feature.

4 Experimental Setup

4.1 Dataset

We used NSURL corpus as training data. It is a Persian NER corpus with more than 900 thousand words that is manually labeled for NER tasks. This corpus was published by NSURL-2019 Workshop for Farsi (Persian) NER Task.

4.2 Evaluation Measure

For Evaluation of NER systems, most of the researches use precision, recall, and F-score as performance measures. Precision is the number of NEs a system correctly detected divided by the total number of NEs identified by the system. Recall is the number of NEs a system correctly detected divided by the total number of NEs contained in the input text. F-Score combines these two into a single score and is defined with the following equation (Tsai et al., 2006).

$$F - score = \frac{2 * precision * recall}{\text{precision} + \text{recall}}$$

5 Experiments and Result

We participated in NER resolution Shared Task for Farsi under the NSURL-2019 Workshop as Team-4. Our results in the workshop are shown in Tables 2 to 6. As we mentioned in section 4, the NSURL corpus is prepared as a training NER corpus but we used only 57 percent of corpus because of the limitations in hardware and computation platform. We expect that the performance of our system be improved if all of the dataset is used for the training phase of the system. To reduce the computational complexity, we removed long sentences with less than two tags. Table 7 shows the characteristics of NSURL and our shortened training corpus.

Corpus	Sentence	Word	Tag
NSURL	23,321	912,032	100,118
Sh_NSURL	10,388	502,989	85,265

Table 2: The characteristics of Corpus.

The results of phrase level and word level NER model for 3 classes (Person, Location and Organization) are shown Table 3 and 4. Moreover, the results in phrase level and word level NER model for 7 classes (Person, Location, Organization, Time, Date, Currency and Percent) are shown in Table 5 and 6.

Although we used only 57 percent of training data, we got acceptable results in NSRUL workshop. In Tables 3 to 5, our results show a lower recall compared to some groups, but we got a better result in precision measure. Table 6 shows the details of phrase level evaluation for 7 classes. As we expected, we got a better result in new tags such as ‘time’, ‘date, ‘money’ and ‘percent’.

Test Data 1	P	R	F1
In Domain	**87.5**	76.0	81.3
Out Domain	**87.5**	76.0	81.3
Total	**86.8**	72.3	78.9

Table 3: Phrase -level evaluation for subtask A: 3-classes

Test Data 1	P	R	F1
In Domain	**90.1**	78.2	83.7
Out Domain	88.7	70.2	78.4
Total	**89.4**	73.5	80.7

Table 4: Word -level evaluation for subtask A: 3-classes

Test Data 1	P	R	F1
In Domain	**87.0**	76.1	81.2
Out Domain	**86.2**	70.2	77.4
Total	**86.5**	72.7	79.0

Table5: Phrase -level evaluation for subtask A: 7-classes

Test Data 1	P	R	F1
In Domain	89.2	83.1	86.1
Out Domain	89.8	76.5	82.6
Total	89.7	79.4	84.2

Table 6: Word-level evaluation for subtask A: 7-classes

Test Data 1	F1
Per	76.2
ORG	75.9
LOC	82.8
DAT	**76.0**
TIM	67.1
MON	**91.3**
PCT	**93.6**
Total F1	79.0

Table 7: Details of phrase-level evaluation for subtask B: 7-classes

6 Conclusion

Supervised methods are domain based so that they generate good results but only on their specific domain. External knowledge can help supervised methods especially if they have common information with test data. In this paper, we extracted useful information from a large unlabeled corpus that it is in the same domain with the test data, both of them are from Persian news, so we added the gazetteer as a new feature to our supervised model. Our results show that this new feature is effective in our named entity recognition model and outperforms the ordinary model.

Acknowledgments

This research is a part of News Dashboard project to be deployed for Islamic Republic of Iran Broadcasting (IRIB). The authors would like to thank all of the members of the above mentioned project. Special credit goes to Dr. Shirin Ghanbari for her warm support.

References

Andrei Mikheev, Marc Moens, and Claire Grover. 1999. *Named entity recognition without gazetteers*. *In Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics, Association for Computational Linguistics pages 1-8.*

Antonio Toral and Monica Monachini. 2008. Named entity wordnet. *In Proceedings of the 6th International Conference on Language Resources and Evaluation.*

Atefeh Zafarian, Ali Rokni, Shahram Khadivi, and Sonia Ghiasifard. 2015. *Semi-supervised learning for named entity recognition using weakly labeled training data. In 2015 The International Symposium on Artificial Intelligence and Signal Processing (AISP), pages 129-135. IEEE.*

Christian Bøhn, and Kjetil Nørvåg. 2010. *Extracting named entities and synonyms from wikipedia. In 2010 24th IEEE International Conference on Advanced Information Networking and Applications, pages 1300-1307. IEEE.*

David Nadeau, Peter D. Turney, and Stan Matwin. 2006. *Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In Conference of the Canadian society for computational studies of intelligence, pages 266-277. Springer, Berlin, Heidelberg.*

Joel Nothman, James R. Curran, and Tara Murphy. 2008. Transforming Wikipedia into named entity training data. *In Proceedings of the Australasian Language Technology Association Workshop 2008, pages 124-132.*

Kentaro Torisawa. 2007. Exploiting Wikipedia as external knowledge for named entity recognition. *In Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL),* pages 698-707.

Nasrin Taghizadeh, Zeinab Borhani-fard, Melika Golestani-Pour, Mojgan Farhoodi, Maryam

Mahmoudi, Masoumeh Azimzadeh and Heshaam Faili. 2019. *Named Entity Recognition (NER) in Farsi. In Proceedings of the first International Workshop on NLP Solutions for Under Resourced Languages, Trento, Italy.*

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. *Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. Artificial intelligence 165(1):91-134.*

Richard Tzong-Han Tsai, Shih-Hung Wu, Wen-Chi Chou, Yu-Chun Lin, Ding He, Jieh Hsiang, Ting-Yi Sung, and Wen-Lian Hsu. *2006. Various criteria in the evaluation of biomedical named entity recognition. BMC bioinformatics 7, no. 1: 92.*

Salar Mohtaj, Behnam Roshanfekr, Atefeh Zafarian, and Habibollah Asghari. 2018. Parsivar: A language processing toolkit for persian. *In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).*

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. *In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pages 708-716.*

The Inception Team at NSURL-2019 Task 8: Semantic Question Similarity in Arabic

Hana Al-Theiabat and Aisha Al-Sadi *
Jordan University of Science and Technology, Irbid, Jordan
haaltheiabat13@cit.just.edu.jo, asalsadi16@cit.just.edu.jo

Abstract

This paper describes our method for the task of Semantic Question Similarity in Arabic in the workshop on NLP Solutions for Under Resourced Languages (NSURL). The aim is to build a model that is able to detect similar semantic questions in Arabic language for the provided dataset. Different methods of determining questions similarity are explored in this work. The proposed models achieved high F1-scores, which range from (88% to 96%). Our official best result is produced from the ensemble model of using pre-trained multilingual BERT model with different random seeds with 95.924% F1-Score, which ranks the first among nine participants teams.

1 Introduction

Semantic matching or semantic similarity is a significant part of natural language processing (NLP) field for its variety of tasks. It used to measure the similarity and the relationship between different textual elements, such as words, sentences, or documents. Semantic matching has been involved in many NLP applications; including question answering, where it is used to assess question answering and retrieval tasks by employing it to estimate the similarity of query answer among all candidate answers (Wang et al., 2016). In addition, it has played a significant role in top-k re-ranking in machine translation (Brown et al., 1993), information extraction (Grishman, 1997) and automatic text summarization (Ponzanelli et al., 2015).

Natural language has complicated structures either from sequential or hierarchical perspectives, capturing the relationship between two questions is becoming a challenging task. For example, questions that have the same meaning while their words have a different order. An effective semantic matching algorithm, therefore, needs to consider an appropriate semantic representation to capture the similarity without being affected with words order.

This paper focuses on detecting semantic question similarity, which is a common challenge in Question-and-answer (Q&A) websites, such as Quora and Stack Overflow. This work targets Arabic questions dataset published by Mawdoo3 AI[1]. Most of these questions are related to information provided by Mawdoo3.com which is the largest comprehensive Arabic content website. For these websites, the benefit of detecting duplicate questions is to improve the efficiency of search engines by being aware of the different paraphrases of the same question.

The rest of paper is organized as follows. Section 2 presents related works. While Section 3 presents some details about the dataset. Section 4, presents the proposed models for solving the semantic similarity in Arabic language task. Results for all proposed models and the final results are presented in Section 5. Finally, the paper conclusion is presented in Section 6.

2 Related Work

Semantic matching has been a long-established problem in NLP. Many approaches were proposed to solve this problem. The conventional approaches were mainly based on representing text as a vector of word features. The bag-of-words (BoW) method (Wu et al., 2008) employed the word occurrence and Term Frequency-Inverse Document Frequency (TF-IDF) (Paltoglou and Thelwall, 2010) as the word feature. However, these types of models disregard word meaning, orders, and even grammar. In contrast, word embedding models such as word2vec (Mikolov et al., 2013) and Glove (Pennington et al., 2014) have been widely used instead

* These authors contributed equally to the work

[1] https://ai.mawdoo3.com/nsurl-2019-task8

of BoW as they can learn distributional semantic representation for words. So based on word embeddings, the Word Movers Distance (WMD) (Kusner et al., 2015) was proposed to measure the dissimilarity between two texts assuming that similar words should have similar vectors. Although WMD can estimate semantic similarity between texts, the order, and interactions between words are excluded.

Recently many deep learning models have been proposed for text matching. A common framework has been adopted is the Siamese architecture (Mueller and Thyagarajan, 2016; Pang et al., 2016; Severyn and Moschitti, 2015; Wang et al., 2017) where the encoder, which can be either Convolutional Neural Network (CNN) or Recurrent Neural Network (RNN), is applied individually on the two input texts, so both texts are encoded into intermediate contextual representations. Then, the matching result is generated by performing a scoring mechanism over contextual representations. Although this framework supports parameter sharing in its network, it purely learns complicated relationships among texts.

Another framework is based on matching aggregation (Wang and Jiang, 2017) which first matches the small units (such as words) of two texts to produce comparison vectors, then these vectors are aggregated and fed into a CNN or RNN for the final classification. This framework improves capturing the interactive features between two texts, but still it limits exploring the matching in only word-word manner.

As the main focus of this paper is to detect semantically equivalent questions, the following is the review of related approaches that were adopted to detect duplicate questions on Quora dataset. As Quora recently published a dataset of 400K labeled questions, massive researches have been proposed on this dataset for question paraphrase identification challenge (qou). One Relevant approach that was proposed for this challenge is the Bilateral Multi-Perspective Matching model (BIMPM) model (Wang et al., 2017) which encodes two questions with a Bidirectional Long Short-Term Memory Network (BiLSTM). Then, a multi-perspective matching in the two directions is applied to both questions, and for each time step, questions are matched using different types of extensive matching. On Quora dataset, the result of this model reached 88.17%. In (Mirakyan et al., 2018), a novel architecture can obtain a high-level understanding of the question pairs through extracting the semantic features using dense interaction tensors (attention) network which called Densely Interactive Inference Network (DIIN). DIIN outperforms BiLSTM on Quora to achieve accuracy of 89.06%. Moreover, Multi-Task Deep Neural Network (MT-DNN) (Liu et al., 2019) achieved competitive performance on several tasks including question paraphrase on Quora with an accuracy of 89.6%. Specifically, MT-DNN Combined multi-task learning and pre-trained bidirectional transformer model for language representation learning.

3 Dataset Description

The dataset used in this task is provided by Mawdoo3 (Seelawi et al., 2019). It is a dataset for questions in Arabic language, it consists of 11,997 labeled question pairs as training data, and 3,715 question pairs as testing data. Label '1' means the question pairs are similar in semantic where label '0' means the opposite. 55% of the training question pairs are with label '0', and 45% are with label '1'. The max length of question 1 is 14 words with an average of 5.7 words per question, while the max length of question 2 is 28 words with an average of 5.3 words per question. Table 1 shows samples from the training dataset.

question1	question2	is_duplicate
ما هي الطرق الصحيحة للاعتناء بالحامل؟	كيف اهتم بطفلي؟	0
ما هي وسائل الاتصالات الحديثة؟	ماذا نعني بوسائل الاتصال الحديثة؟	1
ما طريقة تحضير محشي الكوسا ؟	من طرق تحضير محشي الكوسا؟	1
ما طريقة تحضير حلى الطبقات؟	من طرق تحضير طبقات الكيك؟	0
من الآيات القرآنية عن الراعي والرعية ؟	ما هو تعريف الراعي والرعية ؟	0
أين تقع قارة أوروبا؟	ما هو موقع اليمن؟	0

Table 1: Question samples from Mawdoo3 dataset

The only processing step that was applied to the dataset is to unify countries names, some examples are shown in Table 2.

الأردن	المملكة الأردنية الهاشمية
سوريا	الجمهورية السورية
سنغافوره	سنغافورا

Table 2: Unify countries names example

4 Methodology

In this work, four different deep learning approaches are presented to solve the semantic similarity task, which are RNN based model, CNN based model, multi-head attention based model,

and finally BERT model. In this section, each model is discussed.

4.1 Convolutional Neural Network Model

In NLP field, CNN has shown the ability to extract most informative n-gram features from the input sequence, and then apply the activation on these features (Kim, 2014). Although CNN is known for the applications in the image processing field, it is used here for text classification application.

The proposed model architecture is shown in Figure 1. Firstly, the words are mapped in the dictionary to get a representation for each word. Then each question is fed to three consecutive layers. In each layer, the convolutional layer is applied, followed by activation and then max pooling. Hence, each question's output is a feature representation which is used to get the similarity label by computing the cosine similarity between the two questions features.

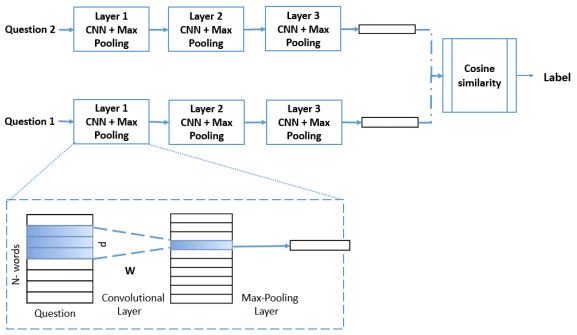


Figure 1: CNN model architecture used for detecting semantic questions similarity

4.2 Recurrent Neural Network Model

The significant advantage of RNNs is the computation of the same task over each element of the sequence, so the output for each block depends on the previous computations. Hence, RNN has been increasingly prevalent in NLP field specifically for RNN types that have a memory to remember the information through the sequence.

In this model, the input is a sequence of question-pairs that are concatenated to represent a single sequence. Then, the sequence is encoded by the dictionary to be fed into a bi-directional Gated Recurrent Units (GRUs) network with 128 hidden units to generate the similarity label as output.

4.3 Multi-head Attention Network Model

Multi-head attention model (Vaswani et al., 2017) allows to learn on various locations of the encoded words. Our network consists of a stacked encoder-decoder structure with eight heads.

For each question-pairs of sequence length n, at each layer l, the encoder maps a sequence of words $Q_l = w_1^l, .., w_n^l$ into hidden representation $h^l = h_1^l, .., h_n^l$. After computing the attention on all positions jointly, the transformer stacks all hidden representation h^l at the current layer l together into matrix H^l. Given h, the decoder then generates output sequence $y^l = y_1^l, .., y_n^l$, and after that apply softmax to estimate the output label. The transformer also contains two sub-layers, a multi-head attention layer, and a position-encoding layer.

The position-encoding layer benefits the network to keep track of relative positions for each word in the sequence since the context and the meaning of a sequence depend on the order of its words.

In the multi-head attention layer, instead of computing single attention on the overall sequence, it jointly gets attention from different representations at different positions. As a result, each head looks differently on encoder output, and the decoder easily learns to retrieve valuable information from the encoder.

4.4 BERT Model

Recently, pre-training language models have shown a significant role to improve many NLP tasks including question-pairs paraphrasing (Dolan and Brockett, 2005). There are two approaches to apply these pre-trained language representations on NLP tasks; either feature-based or fine-tuning. For the feature-based approach (Peters et al., 2018), researchers use the output of pre-trained model as additional features in their models, based on the task they target. On the other hand, the fine-tuning approach (Radford et al., 2018) permits the model to be trained on another task by learning task-specific parameters. The two strategies were mentioned previously have limitations to learning general language representations since they adopt the left-to-right unidirectional architectures. On the other hand, Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018), has strongly outperformed previous cutting-edge unidirectional models.

BERT model relies on the multi-head self-attention mechanism, which enables it to achieve the state-of-the-art accuracy on a wide range of tasks such as, natural language inference, question answering, and sentence classification. The

architecture of BERT model is built upon the transformer layer, which is called the self-attention layer. For each layer, the representations of words are exchanged from previous layers regardless of their positions, in contrast to traditional unidirectional models. For each input word, the model learns bidirectional encoder representations by using the masked language model, which randomly masks some of the words from the input to predict the masked word contextually.

As BERT offers pre-trained models for English language and multilingual model for 104 languages (ber) including the Arabic language, we applied the sentence pairs classification task on Arabic questions through fine-tuning the multilingual model as illustrated in Figure 2.

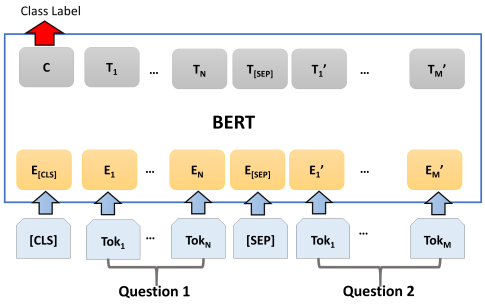


Figure 2: BERT model used for question pair similarity classification task

5 Experiments and Results

For each of the four models explained in the methodology section, different hyper-parameters are used, such as learning rate, number of hidden nodes, and number of epochs. Table 3 shows the main parameters values that give the best results for each model.

The evaluation metric that was used for this task is F1-Score that measures the precision p and recall r together as illustrated in the equations [1]-[3]:

$$F1 = 2p.r/(p + r) \quad (1)$$

$$p = tp/(tp + fp) \quad (2)$$

$$r = tp/(tp + fn) \quad (3)$$

where:

- tp: true positive examples
- fp: false positive examples
- fn: false negative examples

Model	Main parameters
RNN Model	hidden size = 128 cell type = GRU bidirectional = true number of train epochs = 10 train batch size = 512 learning rate = 0.001
CNN Model	number of filters = 50 , 50, 50 filter sizes = 2, 3, 4 number of blocks = 2 number of train epochs = 10 train batch size = 512 learning rate = 0.001
Multi-Head Attention Model	number of heads = 8 use residual = false layer normalization = false number of train epochs = 10 train batch size = 512 learning rate = 0.001
BERT Model	max seq length = 50 train batch size = 8 learning rate = 2e-5 number of train epochs=50

Table 3: Main parameters for each proposed model

Test data evaluation is automatically done online on Kaggle website by submitting the test predictions file. The evaluation system is as the following:

- Public score: calculated with approximately 30% of the data
- Private score: calculated with approximately 70% of the data

During the competition, the public score for each submitted file was shown directly. Then after the competition ended, the submitted file with the highest public score was chosen to calculate its private score and compete other teams based on it.

Table 4 shows the highest F1-Score for each of the four models for the public score and the private score of the test data. As illustrated, BERT model with pre-trained multilingual outperforms the remaining models with F1-score of 96.050% on the public score, and 95.617% on the private score.

Note that the previous results are based on the best public score for every single model of the four models. Since BERT model gives the best results, we conducted other experiments with different random seeds in order to ensemble BERT model. Hard voting is used as ensemble method in which the predictions for each BERT experiments are involved in voting to get the final prediction.

Model	Public Score (%)	Private Score (%)
RNN Model	88.061	88.312
CNN Model	88.330	88.773
Multi-Head Attention Model	86.804	87.889
BERT Model	96.050	95.617

Table 4: Results of 30% of the test data

Model	Public Score (%)	Private Score (%)
Ensemble of best 3 seeds	95.960	96.155
Ensemble of best 4 seeds	96.499	95.924
Ensemble of best 5 seeds	95.691	96.232
Ensemble of best 6 seeds	95.332	96.001

Table 5: BERT Ensemble Results

Table 5 shows the results of the ensemble models of BERT with different number of experiments each with different random seed.

In the ensemble of four and six seeds when the number of votes is equal, high priority was given to the experiments with the best public score.

The result of the ensemble of four seeds has the best public score, so it was chosen for the final evaluation and got the first place. Although other seeds results had lower public scores, they had higher private scores than the official private score. So actually, our best result is 96.232% while the official best result is 95.924%.

6 Conclusion

This paper describes our participation in NSURL Task 8; Semantic Question Similarity in Arabic. Different models were proposed for the task; RNN model, CNN model, Multi-head model, BERT model, and ensemble model of BERT. The ensemble model clearly outperforms all other models in this task by achieving 95.924% F1-Score. This performance ranks first place among nine participating teams.

References

Bert - google research github.

Quora dataset question pairs.

Peter F. Brown, Stephen Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing, IWP@IJCNLP 2005, Jeju Island, Korea, October 2005, 2005*.

Ralph Grishman. 1997. Information extraction: Techniques and challenges. In *Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology, International Summer School, SCIE-97, Frascati, Italy, 14-18, 1997*, pages 10–27.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Improving multi-task deep neural networks via knowledge distillation for natural language understanding. *CoRR*, abs/1904.09482.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Martin Mirakyan, Karen Hambardzumyan, and Hrant Khachatrian. 2018. Natural language inference over interaction space: ICLR 2018 reproducibility report. *CoRR*, abs/1802.03198.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 2786–2792.

Georgios Paltoglou and Mike Thelwall. 2010. A study of information retrieval weighting schemes for sentiment analysis. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 1386–1395.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 2793–2799.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237.

Luca Ponzanelli, Andrea Mocci, and Michele Lanza. 2015. Summarizing complex development artifacts by mining heterogeneous data. In *12th IEEE/ACM Working Conference on Mining Software Repositories, MSR 2015, Florence, Italy, May 16-17, 2015*, pages 401–405.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report, Technical report, OpenAI.

Haitham Seelawi, Ahmad Mustafa, Al-Bataineh Hesham, Wael Farhan, and Hussein T. Al-Natsheh. 2019. NSURL-2019 task 8: Semantic question similarity in arabic. In *Proceedings of the first International Workshop on NLP Solutions for Under Resourced Languages*, NSURL '19, Trento, Italy.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, pages 373–382.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Shuohang Wang and Jing Jiang. 2017. A compare-aggregate model for matching text sequences. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. *arXiv preprint arXiv:1702.03814*.

Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016. Sentence similarity learning by lexical decomposition and composition. *CoRR*, abs/1602.07019.

Ho Chung Wu, Robert Wing Pong Luk, Kam-Fai Wong, and Kui-Lam Kwok. 2008. Interpreting TF-IDF term weights as making relevance decisions. *ACM Trans. Inf. Syst.*, 26(3):13:1–13:37.

Hidden Markov-based Part-of-Speech Tagger for Igbo Language

Iheanetu, Olamma
Department of Computer and Information Science
Covenant University
Ota, Nigeria
olamma.iheanetu@covenantuniversity.edu.ng

Michael, Kingsley
Ecumenical Technology
m.kingsley90@outlook.com

Ojo, Sunday O.
Inclusive African Indigenous Language Technology Institute
Pretoria, South Africa
prof.Sunday.ojo@afrilt.institute

Abstract

Igbo is a resource-scarce Nigerian African language of Bantu language phylum, lacking elctronic linguistic resources in sufficient quantity and quality for the development of human language technologies. Developing Natural Language Processing (NLP) pipeline tools for such a language could be challenging, due to the need to balance the linguistics semantics robustness of the tool with computational parsimony. A Part-of-Speech (POS) tagger is a challenging NLP tool to develop for the language because of its morphological richness poses computational linguistics challenge that could affect the effectiveness of the entire NLP system. In this paper, the experience in developing a POS tagger for the language using the Hidden Markov Model (HMM is presented. It is an on-going project, developed using a small corpus. The results give an approximate accuracy score of 73%, which needs to be improved upon.

1 Introduction

A Part-Of-Speech (POS) tagger is a NLP pipeline tool that inputs text from a source language and assigns a part of speech tag to each word in the text, classifying each as noun, verb, adjective, and so on, or a refinement. POS tagging, sometimes referred to Word Category Disambiguation (WCD) involves giving a word in a text, a unique tag based on the word context and grammatical function. Adjacent and related word to the word of interest plays a huge role in disambiguating the word category, enabling automatic text processing in a language.

POS tags can also be employed for grammatical or lexical pattern searches. In any POS tagging assignment, the aim is to identify the morphosyntactic class of each occurring word based on lexical and contexrual information. Hence, it is possible that in different contexts, a given word may identify with two or more morphosyntactic classes. This scenario informs the importance of engaging human linguistic experts of languages in the inundating task of manually tagging study corpora; an unpopular venture, given the large amounts of data and time needed for such tasks.

For languages where homonyms, especially homographs are prevalent, it becomes pertinent to employ a POS tagger to distinguish between such word occurrences. For example, the word *face* in English could be either a noun or a verb depending on its usage in a sentence. Homonyms also occur in Igbo and are most prevalent especially when diacritics are missing in the text. For example, consider the following homographs in Igbo - *akwa* (cry) [H-H], *àkwà* (bed) [L-L], *àkwa* (egg) [L-H], and *akwà* (cloth) [H-L] all have different meaning in Igbo. The diacritics introduce a measure of distinction and without the diacritics, deciphering the meaning of such words would rely heavily on the context in which they are used, or a POS tagger in the language can be used to disambiguate the words. For homophones, ambiguities must be resolved in order to understand the intended meaning of such words. Otherwise, the ambiguities become misleading especially if a Text-to-Speech (TTS) synthesizer is involved. In reality, most Igbo texts are published without the necessary diacritics due to the unavailability of input tools for such symbols.

Due to the increase in amount of computer readable texts, POS taggers have become very useful and indispensable in computational studies of natural languages. Without automatic text processing like POS tagging, it would require thousands of human hours to manually tag texts, especially when a relatively large corpus is involved. Furthermore, manually tagged corpus is not scalable.

Basically, a POS tagger *learns* from a training set of data that has been manually annotated so that it can automatically tag unseen words appropriately. The tagger also learns the context

in which words are used in order to assign appropriate task. In learning word context, adjacent and related words play a crucial role. POS taggers are as accurate as the training data from which they *learn*.

Part of speech taggers for each language can be mutually unrelated tools and each one can use different tools, techniques, and computational models, as may be dictated by the nuances of the lexical semantic system of each language. Apart from those, there are also multilingual tools which can be trained to process more than one language. The core software stays the same, but a different annotation is used for each language.

2 Related works

2.1 Background to Igbo Language

The Igbo language is one of the Nigerian languages, spoken by the Igbo in South-east Nigeria. The population of Igbo speakers has been put at varying figures by different studies. National Population Commission (2006) estimates an approximate fifteen million Igbo people from the 2006 census; Igbo Open Source project quoted twenty-five million Igbos (http://igbo.sourceforge.net), while Central Intelligence Agency (CIA), U.S.A. (2008) reported a population between twenty-four and twenty-five million Igbos. One to two million other Nigerians speak Igbo as a second language in addition to another three to five million people in Diaspora (Linux, 2010).

Approximately thirty dialects of Igbo exists (UCLA, 2009), some of which are spoken in Abia, Anambra, Enugu, Ebonyi and Imo States, all in the eastern part of Nigeria. Some of these dialects include: Ụmụahịa, Ọnitcha, Orlu, Ngwa, Afikpo, Nsa, Oguta, Aniọcha, Eche, Egbema, Oka (Awka), Bonny-Opobo, Mbaise, Nsụka, Ọhụhụ and Unwana dialects (UCLA, 2009).

Igbo language is a member of the West Benue-Congo languages. Blench and Dendo (2003) formerly classified under the Niger-Congo Kwa language family; a language family that is characterized by high and low tones in which different meanings are applied to the same set of phones (Gale Group, 1999). The language exhibits a rich agglutinative morphology (UCLA, 2009 and Osuagwu, Nwaozuzu, Dike, Nwaogu, and Okoro, 1997). Igbo features a wide variety of highly productive concatenative and non-concatenative morphological processes. Cascaded affixation is a common occurrence in Igbo morphology owing to the agglutinative nature of the language and it is also highly productive in the language.

2.2 Igbo Computational Studies

Due to the increase in the digital textual document and the subtle pressure from the information society to develop human language technologies for computational language studies, Natural Language Processing (NLP) has become indispensable for automatic language processing. The basic resources needed for automatic language processing is computer readable text in a source language. Most resource scarce languages lack this basic requirement and as a result, computational studies of such languages are either slowed down or impeded. Igbo language is among the worlds' less studied languages or resource scarce language because vast electronic linguistic data in the language do not exist.

However, modest efforts have been made in recent times to subject Igbo to computational analysis. Such efforts, as Igbo morphological analyzers (Ayogu, Ignatius, Adetunmbi, Adebayo, Kamelu and Nkiru, 2013), (Iheanetu, and Adeyeye, 2013) and (Iheanetu, 2015), POS tagger (Onyenwe, Onyedinma, Aniegwu and Ezeani, 2019), have recorded some level of successes. Notwithstanding, the language still begs for more efforts towards computational studies in the language.

Iheanetu, (2015) developed an Igbo morphological analyzer using a relatively small corpus and a frequent pattern-based technique. The resulting segmented words had *word label* segmentsinstaed of the conventional syntactic tags. Onyenwe, Hepple, Chinedu and Ezeani (2018) and Onyenwe, Onyedinma, Aniegwu and Ezeani (2019) developed a POS tagger for the language using a modified version of the EAGLE tagset to realize 59 distinct tags. However, they propose the employment of an automatic morphological segmentation in order to realise a more fine-grained tagset for Igbo.

Other recent studies like (Ayogu, Adetunmbi and Ojokoh, 2018) tried to deploy a machine translator for English- Igbo, English-Yoruba and Igbo-Yoruba. On the average, the study was able

to achieve meaningful translation between the languages as depicted by the individual BLEU scores. However, the machine ranslator may need to be improved in order to achive higher BLEU scores with a lower limit of 50.0.

Some generic Open source POS taggers already exist, which boast of their scalability to any language by just re-training the tagger in the source language. These include, the Stanford Log-Linear POS tagger is one of such tools. It was originally developed for French, English, German, Chinese and Arabic languages (https://nlp.stanford.edu/software/tagger.shtml). They also include Python NLTK and Apache OpenNLP. Both of these are machine learning based toolkit for the processing of natural language texts. Theyare most commonly used for NLP tasks, such as tokenization, sentence segmentation, part-of-speech tagging, named entity recognition, chunking, parsing, and co reference resolution. However, their effectiveness in use for any language, is contingent on having robust corpus annotations, that accurately capture the nuances of the lexical semantic system of the language. Here lies the challenge with under-resourced language such as the Igbo language, the paucity of such annotated corpora.

2.3 Ambiguity in igbo POS Tagging

The required parallel corpus for POS tagging is not available for Igbo language, hence, the decision to use a translation of the English Bible, which was translated to Igbo using Google API. The reulting text was not consistent with the Onwu orthography which is the official orthography for Igbo. In addition, the morphology did not fully align with the official Igbo morphology.

Igbo language has a rich agglutinative morphology (UCLA, 2009), which sometimes is expressed in cascades of affixation involving mostly extentional suffixes/ enclitics. Cascaded affixation is a very productive morphological process in Igbo (Iheanetu, 2015). This informs the need to employ morphological segmemntation in achieving a tagset for the language given that some compound words could be read as short phrases. For example consider the Igbo word *abanyekwalarii* meaning “has entered a long time ago”. A morphological segmentation of the word will reveal the morphemes that make up the word.

- *abanyekwalarii*→*a* (Prefix) - *banye* (Verb) – *kwa* (Extensional suffix) – *la* (Enclitic) – *rii* (Enclitic).

Prefixation, suffixation, interfixation, compounding and (root word) modification are the common broad morphological processes in Igbo. As simple as these processes may appear, some of them show a level of complexity, owing to some peculiarities of Igbo language like the concept of *vowel harmony*.

In addition, the high level of agglutination in the language presents some peculiar challenges for POS tagging. The English phrase, ‘must eat completely’ (three words in English) is agglutinatively written *richariri.* where *–ri* is verb root (eat), and *–cha* and *–r*i.*-ri*. are suffixes indicating completion and compulsion respectively.

In the absence of the necessary diacritics, it becomes difficult to differentaite between homonyms. Unfortunately, most Igbo texts are written without these necssary diacritics which are high [ˊ], low [ˋ] tones and downsteps [ˉ] accents for the vowels and syllabic nassals. However in written texts, only the low and midtones are marked (Green and Igwe 1963) in order to facilitate smooth reading and also to make the text wieldy.

3 POS Tagset Design

The Penn Treebank tagset was used for the purposes of this study. In total, the tagset had thirty-six tagsets. However, it was observed that some of the tags did not occur in Igbo (for example, article does not exist in Igbo) while most occuring tags in Igbo were missing. For example, *o* could be used for a personal pronoun *he/ she* or could mean *it,* when it is functioning as an impersonal pronoun. Igbo Particularisers (*nke a* and *nke ahụ*) were not captured in the tags. Therefore, the original Penn Treebank tagset was modified with the addition of the tag IP to capture impersonal pronouns, with the intention of incorporating many others in the future.

3.1 POS Tagging Method

This section discusses the methods and tools used for design and implemetation of the Igbo tagset. However for the alignment, no software was used due to corpus paucity.

3.2 Data Source

This study is on-going, and the results presented here are part of the preliminary results of investigations carried out. Large portions of the text used for this study were sentences from an English Bible [http://bibledatabase.com]. Each verse of the bible in Genesis chapter 1 were tokenized and the tokens were then translated to Igbo using the Google API. Afterwards, the generated Igbo tokens were then manually annotated. In addition to texts from the Igbo Bible, sentences were obtained from twenty newsgroups (http://people.csail.mit.edu/jrennie/20Newsgroups), to accommodate patterns that lace everyday language use. The dataset realized from the outlined sources was relatively small, producing a total of nineteen (19) sentence tokens.
Out of this number, twenty percent (20%) was used as test set (4 sentence tokens) while eighty (80%) was used to train the tagger (15 sentence tokens). The Penn Treebank (Marcus, Santori and Marcinkiewciz, 1993). POS tagset was used for the classification and this tagset includes numbers and punctuations tags. However, we observed that the Penn Treebank tagset did not capture all morphosyntactic classes in Igbo, hence we introduced a morphosyntactic class in the tagset used for the classification. Many more will be likely introduced before the completion of this work. See Table 1.

3.3 The HMM-based Method

The probability-based Hidden Markov Model (HMM); was used for predictive pattern modeling of Igbo POS. HMM is structured to look for the probability of a sequence given an observation:

$$P(S|O) = \frac{P(S,O)}{P(O)}$$

The sequences; (S) represents the tags while the observations (O) represents the sentence tokens.

TABLE I. MODIFIED PENN TREEBANK TAGSET FOR IGBO

No.	Tag	Description
1	CC	Coordinating conjunction
2	CD	Coordinating conjunction
3	DT	Determiner
4	EX	Existential there
5	FW	Foreign word
6	IN	Preposition or subordinating conjunction
7	JJ	Adjective
8	JJR	Adjective, comparative
9	JJS	Adjective, superlative
10	LS	List item marker
11	MD	Modal
12	NN	Noun, singular or mass
13	NNS	Noun, plural
14	NNP	Proper noun, singular
15	NNPS	Proper noun, plural
16	PDT	Predeterminer
17	POS	Possessive ending
18	PRP	Personal pronoun
19	PRPS	Possessive pronoun
20	RB	Adverb
21	RBR	Adverb, comparative
22	RBS	Adverb, superlative
23	RP	Particle
24	SYM	Symbol
25	TO	to
26	UH	Interjection
27	VB	Verb, base form
28	VBD	Verb, past tense
29	VBG	Verb, gerund or present participle
30	VBN	Verb, past participle
31	VBP	Verb, non-3rd person singular present
32	VBZ	Verb, 3rd person singular present
33	WDT	Wh-determiner
34	WP	Wh-pronoun
35	WP$	Possessive wh-pronoun
36	WRB	Wh-adverb
37	IP	Impersonal pronoun

Therefore, the model looks for the best sequences combinations that maximizes P(S|O):
To maximize the probability sequence:

$$P(s_1 \dots s_n | O_1 \dots O_n)$$
$$= P(s_1|s_0)P(o_1|s_1)P(s_2|s_1)P(o_2|s_2) \dots$$

For N observations and K states, there are K^N sequences, and the larger N is the more recursive steps needed in the calculations. Therefore, the use of dynamic programming (shortest path/ tree search algorithms) to arrive at a solution was employed. A Dynamic programming algorithm commonly used with HMM is Viterbi, which attempts to solve the recursive problem:

$$v_1(s_{1=}x) = \max_{k=1}^{L} [v_{i-1}(k).P(x|k).P(o_1|x)]$$

The variable v_i (x) represents the maximum probability that the *i*-th state is x, given that O^i_1 has been seen. At each step, a record of back pointers showing which previous state led to the maximum probability was taken

$$S_{best} = arg \max_s \frac{P(S,O)}{P(O)}$$

4 Evaluation

The training and test set sentence tokens was randomly picked by shuffling the dataset using python script. Accuracy measure was calculated thus:

Accuracy = number of correct tags / number of words

Test accuracy with 13 sentence tokens gave an accuracy of 66.67% .

The error rate was calculated with the formula:

error rate = 1 – Acc

Therefore:

Error rate = 1 – 73.33%

Error_rate = 1 – 0.7333

Error_rate = 0.2667

A demo prototype was put up online (https://igbopos.herokuapp.com/) for further test of the algorithm on new sentence tokens and for dataset gathering for constant upgrade of the performance of the model.

The accuracy of the alignment process had a great impact on the overall accuracy of the tagger. It was observed that some words in the source language (English) were captured by two or more words in the target language (Igbo) and vice versa. Also, the inconsistency with the official Igbo orthography was a major downside of the resulting translation. Some of the words used for translation were either not necessary or was inappropriate. However, the major challenge faced in this study was to manually annotate/ tag the translated Bible verses in order to realize a sufficiently large amount of tags (parallel corpus) to train the Igbo tagger with. Given the short time available for this excercise, it was not possible to realize the desired number of tokens, hence only 19 tokens were used for the alignment and subsequently, to train the tagger. With more tokens and fine-grained tags, it is very possible that the accuracy of the tagger would greatly increase.

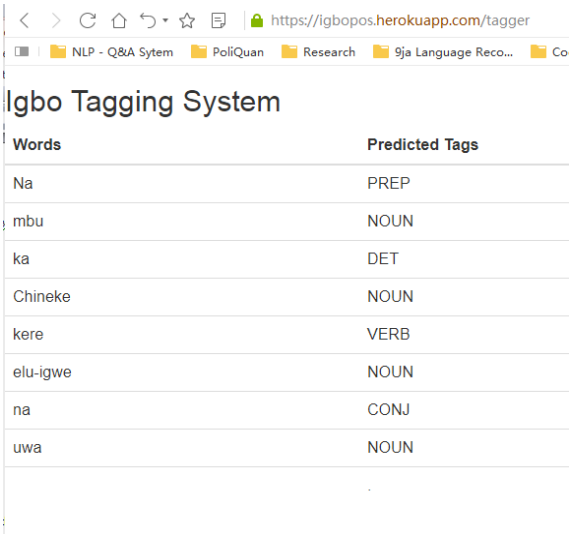


Figure. 1 Screenshot of POS Tagger output

5 Conclusions, Limitations and Futrure works

The study tried to develop an Igbo POS tagger using 19 tokens generated from a corpus consisting the first chapter of the Igbo Bible and a translation of the same using Google API. The resulting translation was not consistent with the official Igbo orthography which is the Onwu orthography and also, sometimes, the accepted morphology of Igbo. The Penn Treebank tagset used did not capture all word forms in Igbo and as such, may need to be modified in order to accommodate the morphological peculiarities of

Igbo. For this study, only one new tag was introduced, among the many that were missing. Hence, more efforts needs to be geared towards achieveing a suitable tagset for training an Igbo POS. A possible direction may be to employ morphological segmentation as suggested by Onyenwe *et al.*,2018.

This is an on-going project and the preliminary test results presented here demonstrate success in the chosen tools for investigation. The researchers hope that adequate amount of data will be generated when the tagger is constantly tested online. In addition, the criticisms will provide a positive feedback for the improvement of the tagger.

References

Ayogu, I., Ignatius, I., Adetunmbi, Adebayo, O., Kamelu and Nkiru, C. 2013. Finite state concatenative morphotactics: the treatment of Igbo verbs. *International Journal of Computing and ICT Research* 7.1:1818-1139.

Ayogu, I., Adetunmbi, A and Ojokoh, B. 2018. Developing Statistical Machine Translation Systemfor English and Nigerian Languages. *Asian Journal of Research in Computer Science. 1(4): 1-8, 2018; Article no.AJRCOS.44217*

Blench, R. and Dendo, M. 2003. Language death in West Africa. Retrieved, June 13, 2015, from www.ling.pdx.edu/childs/DKB.../blench_langauge _d ath_west_africa.pdf

Central Intelligence Agency (CIA), U.S.A. 2008. Igbo people. World Fact Book. Retrieved October 20, 2010, from http://en.wikipedia.org/wiki/CIA_World_Factbook.

Gale Group Inc. 1999. Igbo. Junior Worldmark Encyclopedia of World Cultures. Retrieved August 10, 2010, from Encyclopedia.com: http://www.encyclopedia.com/doc/1G2-3435900354.html

Green, M. and Igwe, E. 1963. A descriptive grammar of Igbo. Berlin: Akademie Verlag.

Igbo Open Source Translation Project. Retrieved October 19, 2010, from http://igbo.sourceforge.net/

Iheanetu, O. and Adeyeye, M. 2013. Finite state representations of reduplication processes in Igbo. *IEEE XPlore Digital Library.* doi:10.1109/AFRCON.2013.6757772. 1-6.

Iheanetu, O. U. 2015. Data-driven model of Igbo morphology. Ph.D thesis. Africa Centre for Information Science (ARCIS), XIV + 208 pages.

Linux, N. 2010. Igbo open source translation project. Sourceforge.net. Retrieved August 10, 2011, from http://igbo.sourceforge.net/

Marcus, M., Santori, B. And Marcinkiewciz, M. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics,* 19:323-330.

National Population Commision. 2006. Retrieved October 18, 2010 from http://www.population.gov.ng/index.php/censuses

Onyenwe, I. E., Hepple, M., Chinedu, U. and Ezeani, I. M. 2018. A basic language resource kitimplementation for the IgboNLP project. ACM Trans, Asian Low-Resour. Lang. Inf. Process., Vol 17, No. 2, Article 10

Onyenwe, I., Onyedinma, E., Aniegwu, G and Ezeani, I. M. 2019. Bootstrapping method for developing part-of-specch tagged corpus in low resource languages tagset - a focus on an African Igbo. *International Journal on Natural Language Computing* (IJNLC), Vol.8, No.1, pp. 13-27.

Osuagwu, B. I. N., Nwaozuzu, G. I., Dike, G. A., Nwaogu, V. N. and Okoro, L. C. 1997. Fundamentals of linguistics. Owerri : Colon Concept Ltd.

University of California, Los Angeles (UCLA) Language Materials Project. 2009. Igbo. UCLA Language Materials Project. Retrieved October 20, 2010, from http://www.lmp.ucla.edu/Profile.aspx?LangID=13&menu=004.

Building Ontology for Yorùbá Language

Okediya Theresa
Department of Computer and Information Science
Covenant University
Ota, Nigeria
theresa.okediya@stu.cu.edu.ng

Afolabi Ibukun
Department of Computer and Information Science
Covenant University
Ota, Nigeria
ibukun.fatudimu@covenantuniversity.edu.ng

Iheanetu Olamma
Department of Computer and Information Science
Covenant University
Ota, Nigeria
olamma.iheanetu@covenantuniversity.edu.ng

Ojo Sunday O.
Inclusive African Indigenous Language Technology Institute
Pretoria, South Africa
prof.Sunday.ojo@afrilt.institute

Abstract

Natural Language Ontology (NLO) provides a formal specification of linguistic semantics knowledge implicit in a natural language. Such a NLO could facilitate a shared understanding of the linguistic semantics system of the language that enhances accuracy of language semantics modelling in Natural Language Processing (NLP). This paper presents the construction of a general purpose ontology for Yorùbá language, one of the under-resourced African languages.Taking as input popular Yorùbá terms obtained from online books, blogs, social websites, and Yorùbá dictionary, the Ontology was constructed, and a prototype implementation made, using the Protégé ontology development tool. Ontology validation and evaluation were done using an automated reasoner.It is emnivisaged that such Yorùbá language ontology will contribute to the development of digital resources for the language, towards its long-term preservation.

1 Introduction

Natural Language Ontology (NLO) provides a formal specification of the most basic categories and relations used in describing a natural language, with the aim of uncovering the ontological categories, notions, and semantic structures that are implicit in the use of the language. It facilitates a shared understanding of the linguistics semantics system of a natural language, and can serve as an input into language modelling to minimize reality-model semantic gap, in Natural Language Processing (NLP). It can also facilitate both the knowledge sharing of annotated linguistic data and the searching of disparate language corpora (Benaissa, Bouchiha, Zouaoui, & Doumi, 2015). Also, in specific terms, an African language, such as Yorùbá, is not only a mirror into the mind of the people group, but also a mirror into their culture and history. Just as they carry their history in their genes, so do they carry same in their language. Hence, the need for a Yorùbá NLO, such as proposed in this paper, is aimed at leveraging the digital development of the under-resourced language. This is towards rendering the language, not only a wider visibility, for its upliftment to academic and scientific status through sound linguistic research.

Due to the increase in the digital textual document, more works have been done and still ongoing to capture the large volume of information that comes from a variety of languages in which only a handful possess the Natural Language Processing (NLP) resources required for developing modern language technologies, researchers have in time made effort to represent different languages such as English, Arabic, French among others (Benaissa et al., 2015; Onyenwe, Hepple, Chinedu, & Ezeani,

2018) but most African languages are still very much under-resourced, one out of the numerous under-resourced languages is Yorùbá which is a language spoken by about thirty-three million people of the South-west, Nigeria(Olúmúyìwá & Aládésanmí, 2017). Yorùbá is believed to have originated from the Igala people about 2000 years ago(Afolabi, Daramola, & Adio, 2014). Out of the 36 states in Nigeria, nine are occupied by Yorùbás which are: Lagos, Ògùn, Ọ̀yọ́, Ọ̀ṣun, Òǹdó, Èkìtì, Kwara, Kogi and Edo States. Across these states, there are different dialects of the language. The dialects are subsumed into five major dialect areas namely: North-West Yorùbá(NWY), North-East Yorùbá(NEY), Central Yorùbá(CY), South-East Yorùbá(SEY) and South-West Yorùbá(SWY). Noteworthy is the fact that this language is spoken worldwide in other nations like Benin, Togo, Ghana, Cote d'Ivoire, Sudan, and Sierra-Leone. Speakers of this language are also in Brazil, Cuba, Haiti, the Caribbean Islands, Trinidad and Tobago, UK and America as well(Ayeomoni, 2012; Olúmúyìwá & Aládésanmí, 2017).

However, like any other African cultural heritage, the Yorùbá language is endangered in the face of inter-ethnic interaction, westernization, and globalization(Hassan, Odéjóbí, & Ògúnfolákàn, 2013). It is therefore of importance to have such a popular language well represented online. Ontology is used to handle information at a semantic level and also play a major role in the semantic web, with this technology, programs, and software agents have access to use the content resources available on the World Wide Web(Lakel & Bendella, 2015), thereby enhancing users' access to information. In view of this study having a well-defined ontology will improve natural language understanding, natural language processing and natural language generation of Yorùbá language.

The aim of this paper is to build a well-defined, lexical ontology for Yorùbá language to be used in NLP system. To achieve this there is need to acquire the knowledge necessary to create the ontology, to identify the concepts to represent, to represent these concepts as classes, to define the conceptual relations and to implement the ontology itself(Bautista-Zambrana, 2015).

The remaining part of this paper is arranged as follows: section 2 gives an overview of the language and related works. Section 3 describes the methodology. Section 4 presents the actual implementation of the work and Section 5 gives the conclusion and recommendation.

2 Related works

Ontologies are used to represent knowledge, an ontology can be used in different fields of knowledge. It can be domain bound which implies the ontology represents knowledge elicited from a specific domain. Different researchers have worked to develop ontologies for different purposes. A domain ontology was developed in (Afolabi et al., 2014) for Nigeria's history, a semi-automated approach was used, the ontology itself was implemented using Protégé software. Similarly,Dramé et al.(2014) proposed a method to construct a bilingual domain ontology, the method uses two approaches: learning ontology from text and reusing existing terminological resources. Rani, Dhar, & Vyas(2017) likewise proposed a modelby exploring two topic modelling algorithms for the purpose of determining the statistical relationship between document and terms and build a topic ontology and ontology graph with little human intervention. Even better, Kethavarapu & Saraswathi(2016) generated data from webpages to build a dynamic ontology using a similarity measure and ontology creation module to generate the Web Ontology Language(owl) file. Also, Alruqimi & Aknin(2019) presented an algorithm for deriving a domain-specific ontology from folksonomy tags, the algorithm takes a domain name as input and produces the corresponding domain ontology as output.

Ontology needs to be evaluated after been created, different evaluation methods have been used in literature, Raad & Cruz(2018) highlighted some evaluation methods which include Gold Standard-based, Corpus-based, Task-based, Criteria based, Structure-based and Complex and Expert based. Lakel & Bendella (2015) proposed a combined approach to improve the process of automatic co-construction of ontologies from a corpus. Expert approach was used to evaluate the ontology in (Dramé et al., 2014; Hassan et al., 2013),Alruqimi & Aknin (2019) used a corpus-based approach, Afolabi et al. (2014) combined the gold standard-based and task-based approach to evaluate the ontology created.

As opposed to conventional ontology, Lexical ontologies are "not based on a specific domain, but they are intended to provide structured

knowledge about lexical issues (words) of a language by linking them to their meanings" (Benaissa et al., 2015).Benaissa et al. (2015) modelled a lexical ontology after the WordNet ontology, the Arabic verb was used as input and Markov Clustering algorithm was used to identify similar verbs. Also,Ishkewy, Harb, & Farahat (2014) developed a software module called Azhary, which is a lexical ontology for Arabic language, the ontology was evaluated using the gold standard-based approach and Arabic WordNet was used as the gold standard. Ontologies have also been constructed for the different domain in Yorùbá, Hassan et al. (2013) described an engineering process of building an ontology for Yorùbá cultural heritage using Formal Concept Analysis for the design, the ontology was implemented with Protégé software and validated using domain experts and ontology experts approach. However there is no lexical ontology for this language yet, hence the reason for this work.

3 Research Framework

3.1 Requirements for the NLO

The major purpose of the ontology is to define the semantic relationship between words in Yorùbá language, which will make information retrieval, automatic text analysis easier and make Yorùbá language available and accessible for digital processing. The architecture of the system is shown in Figure 1. The major use cases of the ontology include:

Knowledge Driven Application: Software that requires knowledge represented in the ontology.

Users: a user interacts with the ontology through a Graphical user interface(GUI) by generating queries. Or a programmer that uses the ontology to create an application using any programming language of choice.

Domain Expert: the domain expert supply the relevant knowledge needed to construct the ontology through their documented materials.

3.2 Data Source

In the cause of this research so far, there was no standardized corpus found for Yorùbá language hence the use of different data sources. Some of the terms were gotten from the Yorùbá dictionary, however only few were selected for the reason that many are no more in use for everyday language. In addition words were retrieved from the internet. Yorùbá words site and some other blogs. The terms were downloaded, saved in Excel spreadsheet and input into the protégé software. The words in the corpus are Yoruba language words. In the language there are seven vowels: [a], [e], [ẹ], [i], [o] [ọ], [u] and four to five nasal vowels: an, ẹn, in, ọn and un. The language has 18 consonants: [b], [d], [f], [g], [gb], [h], [j], [k], [l], [m], [n], [p], [r], [s], [ṣ], [t], [w], and [y](Awoyale, 2008).

Mostly verbs (Ọrọ̀-ìṣe) in Yorùbá language are monosyllabic and monomorphemic examples are wa, lo, je, mu, mu gba and so on while nouns (Ọrọ̀-orúkọ) are polysyllabic and polymorphebic which most times use combinations of the monosyllabic/monomorphemic verbs as stems. Other part of speech represented in Yorùbá language they are: Àpèjúwe (Adjective), Àpólà Àpèjúwe (Adjectival Phrase), Àpọ́nlé (Adverb), Àpólà Àpọ́nlé (Adverbial Phrase), Atọ́kùn (Preposition), Àpólà Atọ́kùn (Prepositional Phrase), Àpólà Ọrọ̀-orúkọ (Noun Phrase), and Àpólà Ọrọ̀-ìṣe (Verb Phrase)

The language is essentially tone-driven which help to deal with Homographs. Take the word "igba" which can mean (plate, two hundred, time, garden egg) it also interesting that unlike some languages, the context of use may not necessarily be used to detect the meaning of a word, take the sentence:

Mu igba wa:
"Bring the plate"
"Bring two hundred"
"Bring the garden egg"

There are three distinct tones used in the language: low, mid and high. Only low (marked with a grave sign) and high (marked with an acute sign) tones are marked on top of the vowel, while the mid tone is left unmarked. "igba" in the sentence above when toned low has only one meaning:

Mu igba wa: "Bring two hundred"

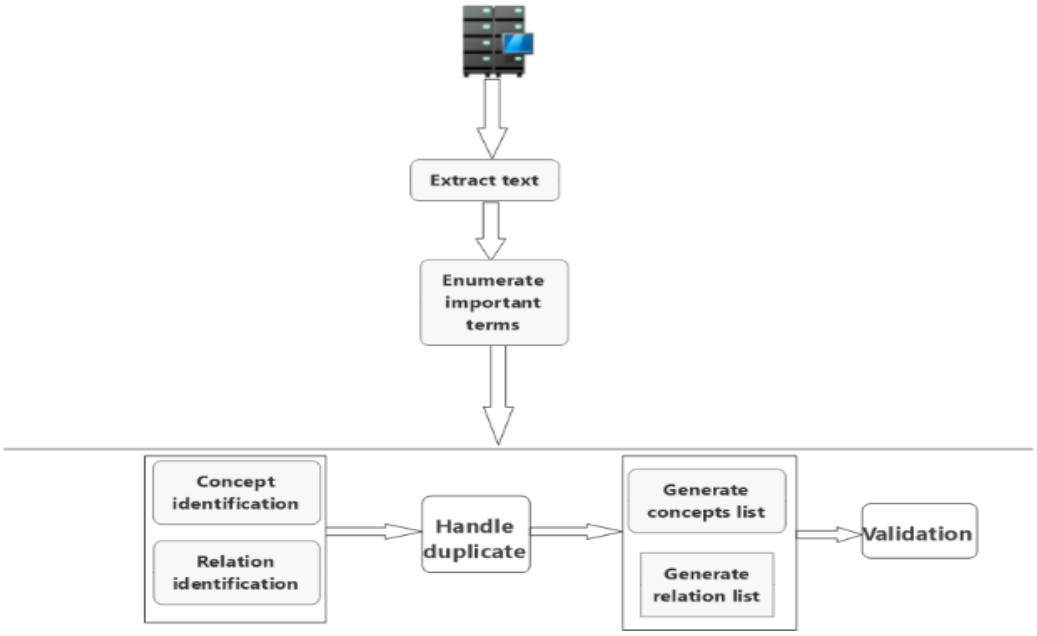


Figure 1: Architecture of the Ontology

3.3 Word and Relation Extraction

The lexical entry to this ontology is the Yorùbá language part of speech, some words can have more than one entries if they have morphological variants such as plural of nouns and inflected form of verb(Staab & Studer, 2009). For example:The verb "wa" which is "come" can have other entries which will point to it as root word, those words won't exist as complete or separate individual, the words include: owa(there is), owa(he came), ewa(telling an elderly person to come, wonwa(they came).

The lexical entries can relate to one another through the following ways(Ishkewy et al., 2014):

- Synonym: B is a synonym of A, if A and B has the same meaning.
- Hypernym: B is a hypernym of A, if A is a (kind of) B.
- Hyponym: B is a hyponym of A, if B is a (kind of) A
- Meronym: B is a meronym of A, if B is a (part of) A
- Holonym: B is a holonym of A, if A is a (part of) B
- Antonym: B is an antonym of A, if A is an (inverse) of B.

3.5 Ontology Building

An ontology development usually encompasses several tasks and **Erreur ! Source du renvoi introuvable.** shows the task in order. Four stages are relevant to the construction of the Yorùbá language ontology, the first is to extract text from different sources as earlier stated in section 1.0, second stage is to identify the concepts and their relations, third phase is to handle duplicates, the exact duplicates(Hassan et al., 2013) are automatically blocked by Protégé while the quasi-exact duplicates and implicit duplicates were manually handled. Finally, validation is done to check for the consistency of the ontology after duplicates have been removed that is to check whether or not all of the statements and definitions in the ontology are mutually consistent. This is achieved using the HermiT reasoner tool in Protégé

4 Implementation and Results

4.1 Protégé OWL Implementation

The ontology implementation was done using Protégé 4.2. There are different ontology languages with different facilities, XML, RDF, RDF(S), OWL and more. However, OWL offers better advantages over others, aside from being the most recent development in standard ontology

languages it also has a richer set of operators - e.g. intersection, union, and negation. It is based on a different logical model which makes it possible for concepts to be defined as well as described.

4.2 Class and Relations description

The concepts were identified from the sources earlier stated, relations were defined across these concepts, and the concepts were arranged hierarchically in a top-down manner as shownin **Erreur ! Source du renvoi introuvable.**that is a more general concept first followed by subclasses. Polysemy deals with relatedness in meaning while Homonymy deals with unrelatedness in meaning. The example below shows homonymous relationship (Babarinde, 2018):

(a) Adé **pa** okùn – 'Ade **sets** rope trap'
(b) Adé **pa** àlọ́ - 'Ade **gives** riddles'
(c) Adé **pa** ìtàn - 'Ade **narrates** a story'
(d) Adé **pa** irọ́ - 'Ade **tells** lies'

Individuals in classes can be related to each other as shown in Figure 2.

4.3 Yorùbá Ontology Validation and Evaluation

According to (Raad & Cruz, 2018), Ontology evaluation is a problem of assessing a given ontology from different perspectives such as accuracy, completeness, conciseness, adaptability, clarity, computational efficiency and consistency. Any evaluation method uses any combination of the criteria earlier listed.

The ontology is compared with Azary, an Arabic
lexical ontology in **Erreur ! Source du renvoi introuvable.**. The ontology constructed was validated using an automated reasoner called HerMiT in Protégé. A reasoner considers the following criteria to assess the performance of an ontology; consistency, satisfiability, and subsumption.

Table 1: NLO and Azhary lexicon

Lexicon	Azhary	YLO
Synonyms relation	Yes	Yes
Hyponym relation	Yes	Yes
Hypernym relation	Yes	Yes
Meronymreation	Yes	Yes
Antonym	Yes	Yes
Happens-before relation	No	Yes
Polysemy	No	Yes
Homonymy	No	Yes

There are different reasoners used to check for the consistency of an ontology but HerMiT does not just determine the consistency of an ontology but can also identify hierarchical relationships between the classes, and much more. The methodology it uses is the hypertableau calculus and it provides the faster way of ontology classification.(Abburu, 2012).
Below is the overall working of the reasoner:

Input : Yorùbá Language Ontology(YLO)

Step1: IF ∃ Model_of_YLO THEN goto step2
ELSE
State = inconsistent

Step2: FOR EACH A in YLO DO
IF ∃ Model_of_YLO SUCH THAT x belongs to A
State = satisfiable
Step3: ∀ class A and B in YLO
Check: IF A IsIn B THEN
State = subsumption

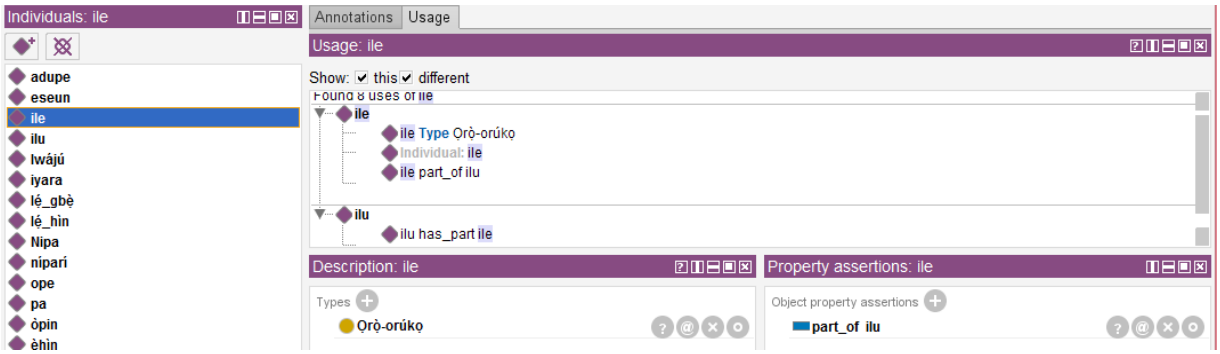


Figure 2: An excerpt of the ontology relative to Homonymy

5 Conclusion and Limitations

The chances for Yorùbá semantic analysis is little since there is no Yorùbá lexical ontology for linguist researchers to depend on, therefore this paper presented the construction of a lexical ontology for the Yorùbá language, using a description logic reasoner the validity of the ontology was tested. The primary use is in automatic text analysis and artificial intelligence applications, it will also support advancement of Natural Language Understanding, Processing and Generation. Moreover, it will make the Yorùbá Language available and accessible for digital processing and sustain the Yorùbá culture in the face of technological advancement. There was no available and well defined corpus for Yorùbá language found so far in the cause of this research which limited the accuracy and consistency of the terms used, also some lexical entries were seen as duplicates because they have the same form as existing ones, this reduced entries.

References

Abburu, S. (2012). A Survey on Ontology Reasoners and Comparison. *International Journal of Computer Applications*, *57*(17), 33–39. https://doi.org/10.5120/9208-3748

Afolabi, I., Daramola, O., & Adio, T. (2014). *Developing Domain Ontology for Nigerian History*. *8*(April), 30–39.

Alruqimi, M., & Aknin, N. (2019). Bridging the Gap between the Social and Semantic Web: Extracting domain-specific ontology from folksonomy. *Journal of King Saud University - Computer and Information Sciences*, *31*(1), 15–21. https://doi.org/https://doi.org/10.1016/j.jksuci.2017.10.005

Amar, F. B. Ben, Gargouri, B., & Hamadou, A. Ben. (2016). Generating core domain ontologies from normalized dictionaries. *Engineering Applications of Artificial Intelligence*, *51*, 230–241. https://doi.org/https://doi.org/10.1016/j.engappai.2016.01.014

Awoyale, Y. (2008). *Global Yoruba Lexical Database v. 1.0*. 1–49.

Ayeomoni, O. M. (2012). A Lexico-syntactic Comparative Analysis of Ondo and Ikale Dialects of the Yoruba Language. *Theory and Practice in Language Studies*, *2*(9), 1802–1810. https://doi.org/10.4304/tpls.2.9.1802-1810

Babarinde, O. (2018). Lexical Ambiguity in Yoruba: its Implications for Second Language Learners. *Journal of Languages, Linguistics and Literary Studies (JOLLS)*, *5*(June), 264–272. Retrieved from http://www.jolls.com.ng

Bautista-Zambrana, M. R. (2015). Creating Corpus-based Ontologies: A Proposal for Preparatory Work. *Procedia - Social and Behavioral Sciences*, *212*, 159–165. https://doi.org/https://doi.org/10.1016/j.sbspro.2015.11.314

Benaissa, B., Bouchiha, D., Zouaoui, A., & Doumi, N. (2015). Building Ontology from Texts. *Procedia Computer Science*, *73*(2015), 7–15. https://doi.org/https://doi.org/10.1016/j.procs.2015.12.042

Bermejo, J. (2007). A simplified guide to create an ontology. *Madrid University*, *DRAFT*, 1–12. Retrieved from http://tierra.aslab.upm.es/documents/controlled/ASLAB-R-2007-004.pdf

Dramé, K., Diallo, G., Delva, F., Dartigues, J. F., Mouillet, E., Salamon, R., & Mougin, F. (2014). Reuse of termino-ontological resources and text corpora for building a multilingual domain

ontology: An application to Alzheimer's disease. *Journal of Biomedical Informatics*, *48*, 171–182. https://doi.org/https://doi.org/10.1016/j.jbi.2013.12.013

Hassan, J. A., Odéjóbí, O. A., & Ògúnfolákàn, B. A. (2013). *Ontology Engineering in Yorùbá Cultural Heritage Domain*. *6*(5).

Ishkewy, H., Harb, H., & Farahat, H. (2014). Azhary: An Arabic Lexical Ontology. In *International journal of Web & Semantic Technology* (Vol. 5). https://doi.org/10.5121/ijwest.2014.5405

Kethavarapu, U. P. K., & Saraswathi, S. (2016). Concept Based Dynamic Ontology Creation for Job Recommendation System. *Procedia Computer Science*, *85*, 915–921. https://doi.org/https://doi.org/10.1016/j.procs.2016.05.282

Lakel, K., & Bendella, F. (2015). Dynamic Evaluation of Ontologies. *Procedia Computer Science*, *73*, 16–23. https://doi.org/https://doi.org/10.1016/j.procs.2015.12.043

Olúmúyìwá, T., & Aládésanmí, Ọmọ́bọ́lá Agnes. (2017). Written Literature in an African Language: An Examination of Interrogative Sentences in Fágúnwà's Novels. *Journal of Siberian Federal University. Humanities & Social Sciences*, *9*(12), 3025–3036. https://doi.org/10.17516/1997-1370-2016-9-12-3025-3036

Onyenwe, I. E., Hepple, M., Chinedu, U., & Ezeani, I. (2018). A Basic Language Resource Kit Implementation for the Igbo NLP Project . *ACM Transactions on Asian and Low-Resource Language Information Processing*, *17*(2), 1–23. https://doi.org/10.1145/3146387

Raad, J., & Cruz, C. (2018). *A Survey on Ontology Evaluation Methods*.

Rani, M., Dhar, A. K., & Vyas, O. P. (2017). Semi-automatic terminology ontology learning based on topic modeling. *Engineering Applications of Artificial Intelligence*, *63*, 108–125. https://doi.org/https://doi.org/10.1016/j.engappai.2017.05.006

Staab, S., & Studer, R. (2009). Handbook on Ontologies, Second edition (International Handbooks on Information Systems) 2009. In *International Handbooks on Information Systems*.