Deep learning based Semantic Textual Similarity for Applications in Translation Technology

Tharindu Ranasinghe

A thesis submitted in partial fulfilment of the requirements of the University of Wolverhampton for the degree of Doctor of Philosophy

2021

This work or any part thereof has not previously been presented in any form to the University or to any other body whether for the purposes of assessment, publication or for any other purpose (unless otherwise indicated). Save for any express acknowledgements, references and/or bibliographies cited in the work, I confirm that the intellectual content of the work is the result of my own efforts and of no other person.

Signature: . . . .

Date: . . . .

Abstract

Acknowledgements

Contents

List of Tables

List of Figures

Listings

Introduction

Part I

Semantic Textual Similarity

Chapter 1

Introduction

1.1 Semantic Textual Similarity Approaches

Over the years, researchers have proposed numerous STS methods. Most of the early approaches were based on traditional machine learning and involved heavy feature engineering [1]. With the advances of word embeddings, and as a result of the success neural networks have achieved in other fields, most of the methods proposed in recent years rely on neural architectures [2, 3]. Neural networks are preferred over traditional machine learning models as they generally tend to perform better than traditional machine learning models. They also do not rely on explicit linguistics features which have to be extracted before the ML model is learnt. Determining the best linguistic features for calculating STS is not an easy task as it requires a good understanding of the linguistic phenomenon and relies on researchers' intuition. In addition, calculating these features is usually not an easy task, especially for languages other than English. Therefore, in contrast to traditional ML methods, models based on word embeddings and neural networks can be easily applied to other languages.

As stated in the Chapter the machine learning algorithms we experimented can be classified in to two min categories: Unsupervised STS methods and Supervised STS methods. In the Chapter 2 we evaluate the current STS state of the arts methods that uses word embeddings and we improve state of the arts STS methods using contextual embeddings.

In Chapter 3 we explore another unsupervised STS method using sentence encoders. We use three different sentence encoders analyse their performance in various aspects of English STS and also evaluate their portability to different languages and domains.

Siamese Neural Networks are a special kind of neural network that are being used commonly in STS tasks. It is a supervised STS method which we discuss comprehensively in Chapter 4. We evaluate the existing Siamese Neural Network architectures in STS datasets and propose a novel Siamese Neural Network architecture, MAGRU: an efficient and more accurate Siamese Neural Network architecture for STS tasks. We also asses its performance on different languages and different domains.

In the final chapter of the Part I of this thesis, we explore the newly released transformers in STS tasks. We bring together various transformer architectures like BERT [4], XLNet [5], RoBERTa [6] etc and investigate their performance in various STS datasets in Chapter 5.

The remainder of this chapter is structured as follows. Section 1.2 discuss the various datasets we used in *"Semantic Textual Similarity"* part of the thesis. We also briefly analyse the datasets for common properties. In the Section 1.4 we discuss the main contributions we have to the community with the *"Semantic Textual Similarity"* part of the thesis. The chapter concludes with the conclusions.

1.2 Datasets

We experimented with several datasets throughout the experiments in the Semantic Textual Similarity Section. In order to maintain the versatility of our methods we experimented with several English datasets as well as several non English datasets and several datasets from different domains which we will introduce in this section. All of the datasets which are described here re publicly available and can be considered as STS benchmarks.

1.2.1 English Datasets

1. **SICK dataset** [1] - The SICK data contains 9927 sentence pairs with a 5,000/4,927 training/test split which were employed in the SemEval 2014 Task1: Evaluation of Compositional Distributional Semantic Models on Full Sentences through Semantic Relatedness and Textual Entailment [7]. The dataset has two types of annotations: Semantic Relatedness and Textual Entailment. We only use Semantic Relatedness annotations in our research. SICK was built starting from two existing datasets: the 8K ImageFlickr data set [2] [8] and the SemEval-2012 STS MSR-Video Descriptions dataset [3] [9]. The 8K ImageFlickr dataset is a dataset of images, where each image is associated with five descriptions. To derive SICK sentence pairs the organisers randomly selected 750 images and sampled two descriptions from each of

[1]The SICK dataset is available to download at https://wiki.cimec.unitn.it/tiki-index.php?page=CLIC

[2]The 8K ImageFlickr data set is available at http://hockenmaier.cs.illinois.edu/8k-pictures.html

[3]The SemEval-2012 STS MSR-Video Descriptions dataset is available at https://www.cs.york.ac.uk/semeval-2012/task6/index.html

them. The SemEval2012 STS MSR-Video Descriptions data set is a collection of sentence pairs sampled from the short video snippets which compose the Microsoft Research Video Description Corpus [4]. A subset of 750 sentence pairs have been randomly chosen from this data set to be used in SICK.

In order to generate SICK data from the 1,500 sentence pairs taken from the source data sets, a 3-step process has been applied to each sentence composing the pair, namely *(i) normalisation, (ii) expansion and (iii) pairing* [7]. The *normalisation* step has been carried out on the original sentences to exclude or simplify instances that contained lexical, syntactic or semantic phenomena such as named entities, dates, numbers, multiword expressions etc. In the *expansion* step syntactic and lexical transformations with predictable effects have been applied to each normalized sentence, in order to obtain *(i)* a sentence with a similar meaning, *(ii)* a sentence with a logically contradictory or at least highly contrasting meaning, and *(iii)* a sentence that contains most of the same lexical items, but has a different meaning. Finally, in the *pairing* step each normalised sentence in the pair has been combined with all the sentences resulting from the expansion phase and with the other normalised sentence in the pair. Furthermore, a number of pairs composed of completely unrelated sentences have been added to the data set by randomly taking two sentences from two different pairs [7].

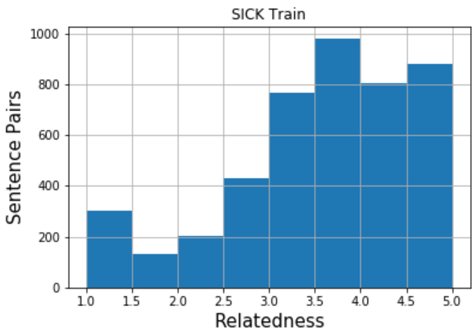
[4] The Microsoft Research Video Description Corpus is available to download at `https://research.microsoft.com/en-us/downloads/38cf15fd-b8df-477e-a4e4-a4680caa75af/`

Each pair in the SICK dataset has been annotated to mark the degree to which the two sentence meanings are related (on a 5-point scale). The ratings have been collected through a large crowdsourcing study, where each pair has been evaluated by 10 different annotators. Once all the annotations were collected, the relatedness gold score has been computed for each pair as the average of the ten ratings assigned by the annotators [7]. Table 1.1 shows examples of sentence pairs with different degrees of semantic relatedness; gold relatedness scores are expressed on a 5-point rating scale. Given a test sentence pair the machine learning models require to predict a value between 0-5 which reflects the relatedness of the given sentence pair.

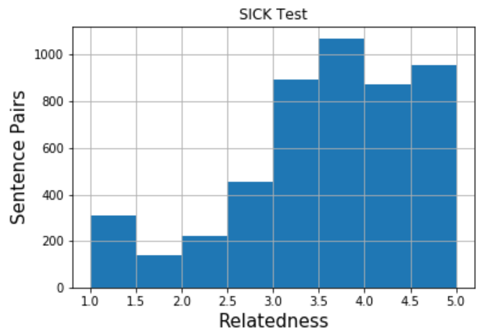
Sentence Pair	**Relatedness**
1. A little girl is looking at a woman in costume. 2. A young girl is looking at a woman in costume.	4.7
1. Nobody is pouring ingredients into a pot. 2. Someone is pouring ingredients into a pot.	3.5
1. Someone is pouring ingredients into a pot. 2. A man is removing vegetables from a pot.	2.8
1. A man is jumping into an empty pool. 2. There is no biker jumping in the air.	1.6

Table 1.1: Example sentence pairs from the SICK dataset with their gold relatedness scores (on a 5-point rating scale). **Sentence Pair** column shows the two sentence and **Relatedness** column denotes the annotated relatedness score.

Figure 1.1 shows the distribution of the relatedness value in SICK training and SICK testing set. It is clear that there are more sentence pairs with a high relatedness values compared to low relatedness values. SICK train



(a) Relatedness distribution of SICK training test



(b) Relatedness distribution of SICK testing set

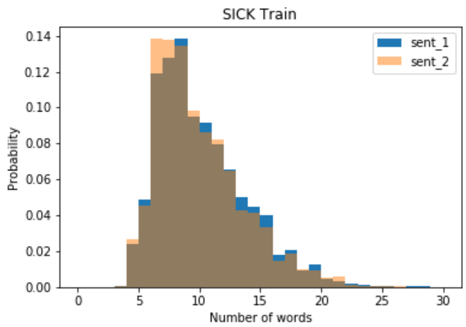
Figure 1.1: Relatedness distribution of SICK train and SICK test. *Sentence Pairs* shows the number of sentence pairs that a certain *Relatedness bin* has.

Measure	SICK Train		SICK Test	
	Sent_1	**Sent_2**	**Sent_1**	**Sent_2**
Word Count Mean	9.73	9.52	9.69	9.53
Word Count STD	3.66	3.70	3.69	3.65
Word Count MAX	28	32	28	30
Word Count MIN	3	3	3	3

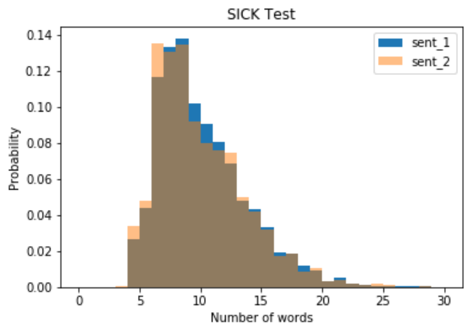
Table 1.2: Word count stats in SICK training and SICK testing. *STD* indicates the standard deviation and the other acronyms indicate the common meaning

and SICK test follows a similar distribution.

In Figure 1.2 we visualise the normalised distribution of word count for both sentence 1 and sentence 2 in SICK train and SICK test. Both sentences have a similar distribution reaching the maximum around 9 words. SICK train and SICK test follows a similar pattern in word count distribution too. Additionally we show some word count statistics in Table 1.2. In SICK train number of words for a sentence ranges from 3 to 32 and have the mean number of words around 9.5. These statistics are extremely close in



(a) Normalised distribution of word count in SICK train



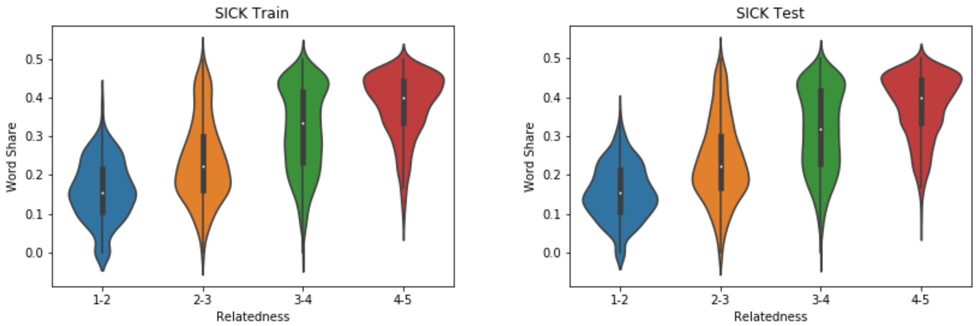
(b) Normalised distribution of word count in SICK test

Figure 1.2: Normalised distribution of word count in SICK train and SICK test. *Number of words* indicates the word count and *Probability* shows the total probability of a sentence with that word count appearing in the dataset.

SICK test too.

The common judgement in STS is that, when two sentences share a large number of words, the relatedness of that two sentences should be higher. In fact, in early feature based approaches of calculating semantic textual similarity, the number of overlapping words between the two sentences was a common feature [10, 11, 12, 13]. Systems like Vilariño et al. [10], Lynum et al. [12] use the number of words common in two sentences as a feature directly while systems like Gupta et al. [11], Chávez et al. [13] use Jaccard Similarity Coefficient as a feature, which is a measurement based on word overlap. To observe, whether the number of words common in the two sentences has a relationship on the relatedness, we draw a violin plot [5] for each relatedness score bins with word share in Figure 1.3.

[5]Violin plots are similar to box plots, except that they also show the probability density of the data at different values, usually smoothed by a kernel density estimator.



(a) Word share against relatedness bins in SICK train

(b) Word share against relatedness bins in SICK test

Figure 1.3: Word share against relatedness bins in SICK train and SICK test. *Word Share* indicates the ratio between number of common words in the two sentences to total number of words in the two sentences against each *Relatedness* bins

In figure 1.3, it is clear that sentence pairs with a higher relatedness tend to have a high word share. However, it should be noted that, in the "2-3" relatedness score bin, there are some sentence pairs with a high word share. Most common example for such a case would be sentence 2 is the complete negation of the sentence 1. In such cases the two sentences share a large potion of the words and one sentence have the *"not"* word that gives a complete opposite meaning compared to the other sentence. Similarly "4-5" relatedness score bin has some sentence pairs with a low word share. Those sentence pairs does not contain the same words but will be having synonyms and possess the same overall meaning. Therefore, the STS methods that focusses on word share won't perform well in SICK dataset.

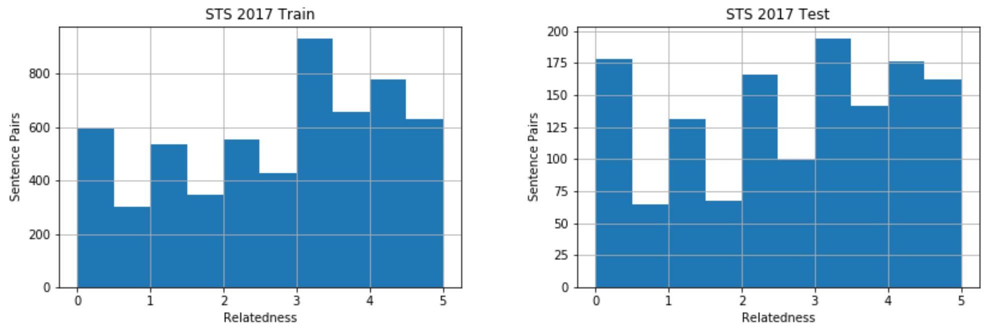
A clear strength in the SICK dataset is that training set and the testing set reflects similar properties so that a properly trained machine learning model on SICK train should give good results to the SICK test set as well.

2. **STS 2017 English Dataset** [6] The second English STS dataset we used to experiment in this section is STS 2017 English Dataset which was employed in SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Cross-lingual Focused Evaluation which is the most recent STS task in SemEval [14]. As the training data for the competition, participants were encouraged to make use of all existing data sets from prior STS evaluations including all previously released trial, training and evaluation data from SemEval 2012 - 2016 [9, 15, 16, 17, 18]. Once combined we had 8277 sentence pairs for training. More information about the datasets used to build the training set is available in Table 1.3.

   On the other hand, a fresh test set of 250 sentence pairs was provided by SemEval-2017 STS Task organisers [14]. The Stanford Natural Language Inference (SNLI) corpus [19] was the primary data source for this test set. Similar to the SICK dataset, Each pair in the STS 2017 English Test set has been annotated to mark the degree to which the two sentence meanings are related (on a 5-point scale). The ratings have been collected through crowdsourcing on Amazon Mechanical Turk[7]. Five annotations have been collected per pair and gold score has been computed for each pair as the average of the five ratings assigned by the annotators. However, unlike the

[6]The STS 2017 English Dataset is available to download at http://ixa2.si.ehu.es/stswiki/

[7]Amazon Mechanical Turk is a crowdsourcing website for businesses to hire remotely located *crowd workers* to perform discrete on-demand tasks. It is available at https://www.mturk.com/



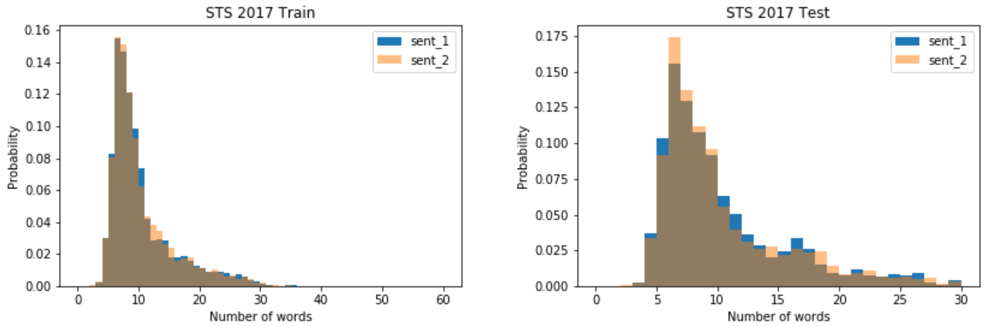
(a) Relatedness distribution of STS 2017 training test

(b) Relatedness distribution of STS 2017 testing set

Figure 1.4: Relatedness distribution of STS 2017 train and STS 2017 test. *Sentence Pairs* shows the number of sentence pairs that a certain *Relatedness bin* has.

SICK dataset, the organisers has a clear explanations for the score ranges. Table 1.4 shows some example sentence pairs from the dataset with the gold labels and their explanations. Similar to the SICK dataset, the machine learning models require to predict a value between 0-5 which reflects the similarity of the given sentence pair.

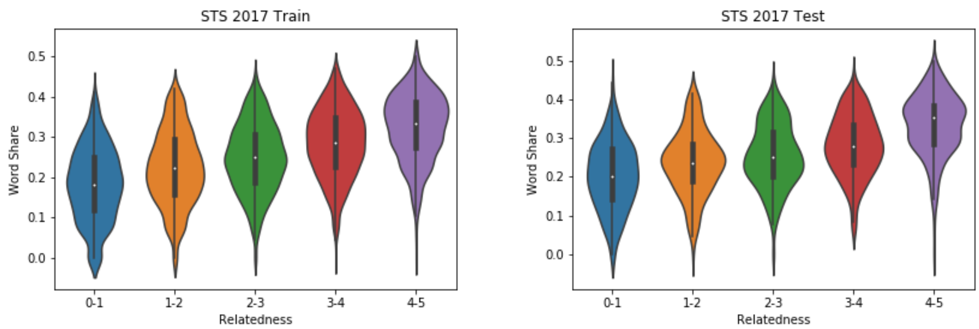
Similar to the SICK dataset, we calculate some statistics and produce some graphs. Figure 1.4 shows the relatedness distribution and Figure 1.5 shows the normalised distribution of word count for sentence 1 and sentence 2 in STS 2017 train and test sets. Most of these statistics are similar to the SICK dataset. One notable change is the maximum word count in STS 2017 training dataset which is 57 in sentence 1 and 48 in sentence 2 according to Table 1.5 while both SICK datasets' and STS 2017 test set's maximum word count is limited to 30. We believe that the reason is STS train is composed with many sources including news articles which can have lengthy



(a) Normalised distribution of word count in STS 2017 train

(b) Normalised distribution of word count in STS 2017 test

Figure 1.5: Normalised distribution of word count in STS 2017 train and STS 2017 test. *Number of words* indicates the word count and *Probability* shows the total probability of a sentence with that word count appearing in the dataset.



(a) Word share against relatedness bins in STS 2017 train

(b) Word share against relatedness bins in STS 2017 test

Figure 1.6: Word share against relatedness bins in STS 2017 train and STS 2017 test. *Word Share* indicates the ratio between number of common words in the two sentences to total number of words in the two sentences against each *Relatedness* bins

Year	Dataset	Pairs	Source
2012 [9]	MSRpar	1500	newswire
	MSRvid	1500	videos
	OnWN	750	glosses
	SMTnews	750	WMT eval.
	SMTeuroparl	750	WMT eval.
2013 [15]	HDL	750	newswire
	FNWN	189	glosses
	OnWN	561	glosses
	SMT	750	MT eval.
2014 [16]	HDL	750	newswire headlines
	OnWN	750	glosses
	Deft-forum	450	forum posts
	Deft-news	300	news summary
	Images	750	image descriptions
	Tweet-news	750	tweet-news pairs
2015 [17]	HDL	750	newswire headlines
	Images	750	image descriptions
	Ans.-student	750	student answers
	Ans.-forum	375	Q&A forum answers
	Belief	375	committed belief
2016 [18]	HDL	249	newswire headlines
	Plagiarism	230	short-answer plag.
	post-editing	244	MT postedits
	Ans.-Ans.	254	Q&A forum answers
	Quest.-Quest.	209	Q&A forum questions
2017 [14]	Trial	23	Mixed STS 2016

Table 1.3: Information about the datasets used to build the English STS 2017 training set. The **Year** column shows the year of the SemEval competition that the dataset got released. **Dataset** column expresses the acronym used describe a dataset in that year. **Pairs** is the number of sentence pairs in that particular dataset and **Source** shows the source of the sentence pairs.

sentences. However, the STS algorithm should be able to properly handle this imbalance nature between STS 2017 train and test set.

In Figure 1.6 we draw a violin plot for each relatedness score bin with

Sentence Pair	Relatedness
The two sentences are completely equivalent as they mean the same thing. 1. The bird is bathing in the sink. 2. Birdie is washing itself in the water basin.	5
The two sentences are completely equivalent as they mean the same thing. 1. The bird is bathing in the sink. 2. Birdie is washing itself in the water basin.	4
The two sentences are roughly equivalent, but some important information differs/missing. 1. John said he is considered a witness but not a suspect. 2. "He is not a suspect anymore." John said.	3
The two sentences are not equivalent, but share some details. 1. They flew out of the nest in groups. 2. They flew into the nest together.	2
The two sentences are not equivalent, but are on the same topic. 1. The woman is playing the violin. 2. The young lady enjoys listening to the guitar.	1
The two sentences are completely dissimilar 1. The black dog is running through the snow. 2. A race car driver is driving his car through the mud.	0

Table 1.4: Example sentence pairs from the STS2017 English dataset with their gold relatedness scores (on a 5-point rating scale) and explanations. **Sentence Pair** column shows the two sentence and **Relatedness** column denotes the annotated relatedness score.

word share. We can see that generally higher word share leads to higher relatedness, but still there can be sentence pairs contradicts this which is similar to the observation we had with SICK dataset.

Since the statics of SICK and STS 2017 datasets are similar one dataset can be used to augment the training data in the other dataset which can lead to

Measure	STS 2017 Train		STS 2017 Test	
	Sent_1	**Sent_2**	**Sent_1**	**Sent_2**
Word Count Mean	10.01	9.94	9.83	9.80
Word Count STD	5.52	5.36	5.14	5.14
Word Count MAX	57	48	30	30
Word Count MIN	3	2	3	2

Table 1.5: Word count stats in STS 2017 training and STS 2017 testing.*STD* indicates the standard deviation and the other acronyms indicate the common meaning

better results as neural networks perform better with more data [20, 21]. We hope to experiment this with supervised machine learning models in Chapters 4 and 5.

3. **Quora Question Pairs** [8] The Quora Question Pairs dataset is a big dataset which was first released for a Kaggle Competition[9]. Quora is a question-and-answer website where questions are asked, answered, followed, and edited by internet users, either factually or in the form of opinions. If a particular new question has been asked before, users merge the new question to the original question flagging it as a duplicate. The organisers used this functionality to create the dataset and did not use a separate annotation process. Their original sampling method has returned an imbalanced dataset with many more true examples of duplicate pairs than non-duplicates. Therefore, the organisers have supplemented the dataset with negative examples. One source of negative examples have been pairs of

[8] The Quora Question Pairs Dataset is available to download at http://qim.fs.quoracdn.net/quora_duplicate_questions.tsv

[9] Kaggle is an online community of data scientists and machine learning practitioners that hosts machine learning competitions. The Quora Question Pairs competition is available on https://www.kaggle.com/c/quora-question-pairs

related question which, although pertaining to similar topics, are not truly semantically equivalent.

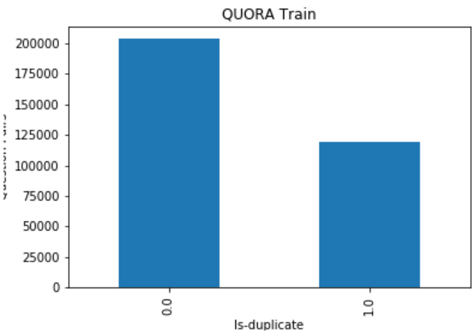
The dataset has 400,000 question pairs and we used 4:1 split on that to separate it into a training set and a test set resulting 320,000 questions pairs in the training set and 80,000 sentence pairs in the testing set. The machine learning models need to predict a value between 0 and 1 that reflects whether it is a duplicate question pair or not. 1 indicates that a certain question pair is a duplicate and 0 indicates it is not a duplicate.

Question Pair	**is-duplicate**
1. What are natural numbers? 2. What is a least natural number?	0
1. Which Pizzas are most popularly ordered in Dominos menu? 2. How many calories does a Dominos Pizza have?	0
1. How do you start a bakery? 2. How can one start a bakery business?	1
1. Should I learn Python or Java first? 2. If I had to choose between learning Java and Python what should I choose to learn first?	1

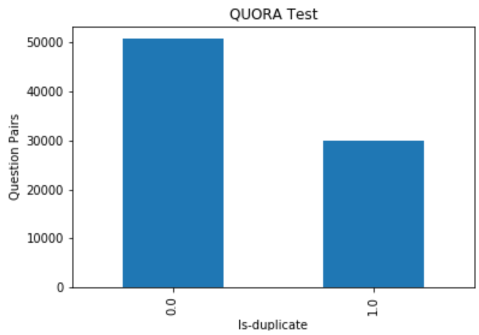
Table 1.6: Example question pairs from the Quora Question Pairs dataset with their gold is-duplicate value. **Question Pair** column shows the two questions and **is-duplicated** column denotes whether it is a duplicated pair or not.

This is different to the previous datasets since it is not artificially created and use day to day language. Since it has more than 300,000 training instances deep learning systems will benefit more when used on this dataset.

In Figure 1.7 we show the distribution of the two classes in QUORA dataset.

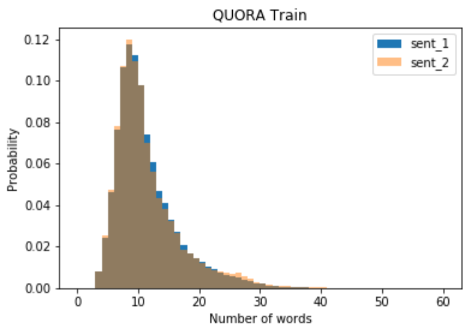


(a) Is-duplicate distribution of QUORA training test

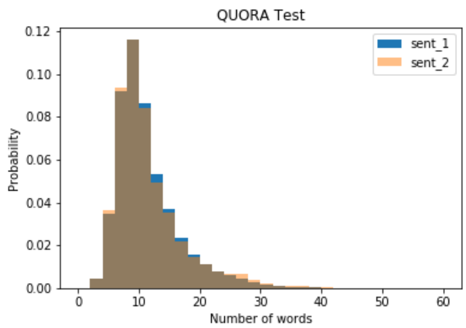


(b) Is-duplicate distribution of QUORA testing set

Figure 1.7: Is-duplicate distribution of QUORA train and QUORA test. *Sentence Pairs* shows the number of sentence pairs that a certain *Is-duplicate* has.



(a) Normalised distribution of word count in QUORA train

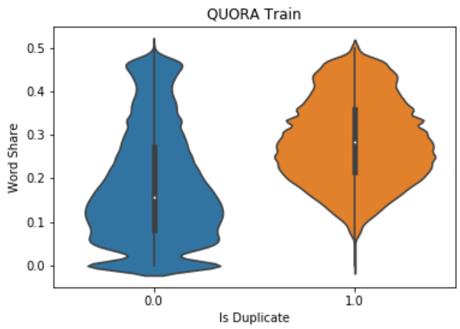


(b) Normalised distribution of word count in QUORA test

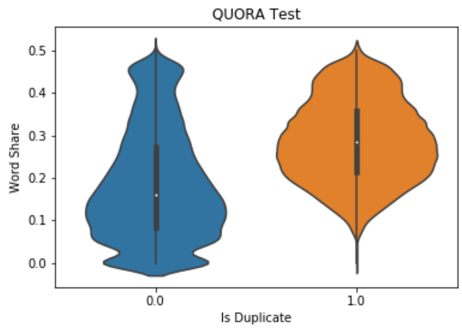
Figure 1.8: Normalised distribution of word count in QUORA train and QUORA test. *Number of words* indicates the word count and *Probability* shows the total probability of a sentence with that word count appearing in the dataset.

Measure	QUORA Train		QUORA Test	
	Ques_1	**Ques_2**	**Ques_1**	**Ques_2**
Word Count Mean	10.95	11.20	10.92	11.14
Word Count STD	5.44	6.31	5.40	6.31
Word Count MAX	125	237	73	237
Word Count MIN	1	1	1	1

Table 1.7: Word count stats in QUORA training and QUORA testing.*STD* indicates the standard deviation and the other acronyms indicate the common meaning



(a) Word share against is-relatedness value in QUORA train



(b) Word share against is-relatedness value in QUORA test

Figure 1.9: Word share against Is-duplicate values in QUORA train and QUORA test. *Word Share* indicates the ratio between number of common words in the two sentences to total number of words in the two sentences against each *Is-duplicate*

The dataset seems to have more non duplicate question pairs than duplicate sentence pairs which is similar to the real world scenario. According to the word count distribution in Figure 1.8 and word count statistics in Table 1.7, it is clear that QUORA datasets contains longer texts than SICK and STS 2017 datasets. Therefore, QUORA dataset should be able to test machine learning models'ability to handle lengthy texts properly.

In Figure 1.9 we show a violin plot for each *"is-duplicate"* value with word share. We can see that duplicate questions have a high word share. However, it should be noted that there are non duplicate question pairs that still have a high word share. The machine learning algorithm should be able to handle them properly.

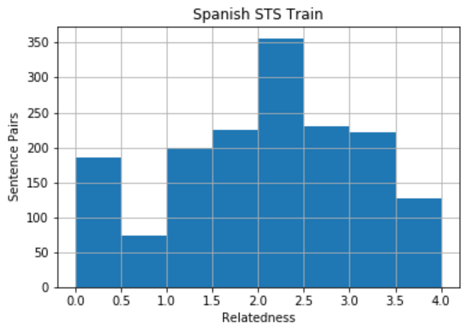
According to statistics provided by the Director of Product Management at Quora on 17 September 2018, over 100 million people visit Quora every month, which raises the problem of different users asking similar questions with same intent but in different words [22]. Multiple questions with the same intent can cause seekers to spend more time finding the best answer to their question, and make writers feel they need to answer multiple versions of the same question. Therefore, identifying duplicate questions will make it easier to find high quality answers to questions resulting in an improved experience for Quora writers, seekers, and readers.

1.2.2 Datasets on Other Languages

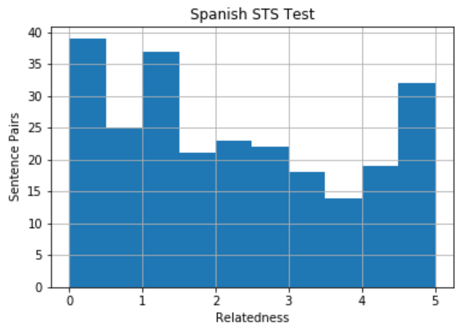
One of the main requirements in our research was to build a STS method without depending on the language. Therefore through out our research we worked on several datasets from different languages. Those non-English datasets are described below.

1. **Spanish STS Dataset** [10] - Spanish STS dataset that we used was employed for Spanish STS subtask in SemEval 2017 Task 1: Semantic Textual Similarity Multilingual and Cross-lingual Focused Evaluation [14]. The training set has 1250 sentence pairs annotated with a relatedness score between 0 and 4. The training set combined several datasets from previous SemEval STS shared tasks also[14]. Table 1.8 shows more information about the training set. There were two sources for test set - Spanish news and Spanish Wikipedia dump having 500 and 250 sentence pairs respectively [14]. Both datasets were annotated with a relatedness score between 0 and 5.

[10]The Spanish STS dataset can be downloaded at http://alt.qcri.org/semeval2017/task1/index.php?id=data-and-tools



(a) Relatedness distribution of Spanish STS training test



(b) Relatedness distribution of Spanish STS testing set

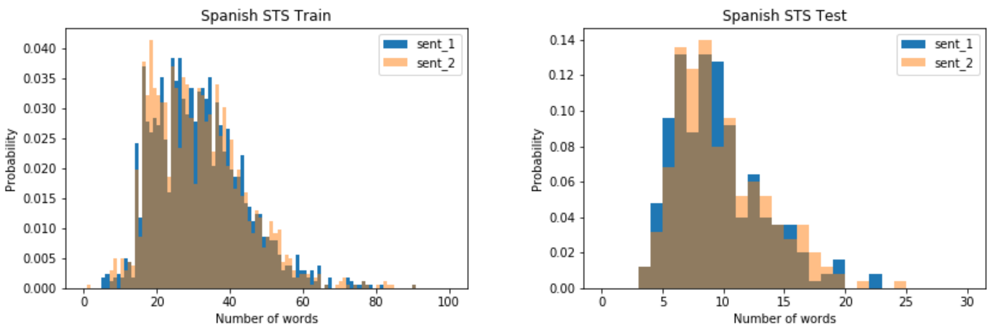
Figure 1.10: Relatedness distribution of Spanish STS train and Spanish STS test. *Sentence Pairs* shows the number of sentence pairs that a certain *Relatedness bin* has.

Table 1.9 shows few pairs of sentences with their similarity score. The machine learning models require to predict a value between 0-5 which reflects the similarity of the given Spanish sentence pair.

Year	Dataset	Pairs	Source
	Trial	56	NR
2014 [16]	Wiki	324	Spanish Wikipedia
	News	480	Newswire
2015 [16]	Wiki	251	Spanish Wikipedia
	News	500	Sewswire

Table 1.8: Information about the datasets used to build the Spanish STS training set. The **Year** column shows the year of the SemEval competition that the dataset got released. **Dataset** column expresses the acronym used describe a dataset in that year. **Pairs** is the number of sentence pairs in that particular dataset and **Source** shows the source of the sentence pairs.

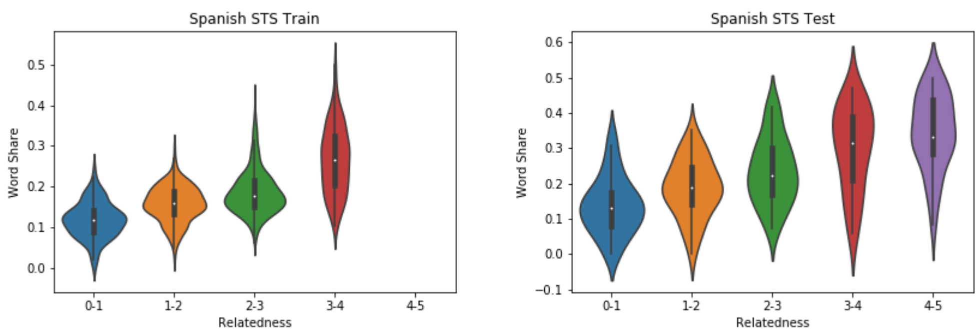
Similar to the English datasets we calculate some statistics and produce some graphs. A key challenge in the Spanish STS dataset is that test set



(a) Normalised distribution of word count in Spanish STS train

(b) Normalised distribution of word count in Spanish STS test

Figure 1.11: Normalised distribution of word count in Spanish STS train and Spanish STS test. *Number of words* indicates the word count and *Probability* shows the total probability of a sentence with that word count appearing in the dataset.



(a) Word share against relatedness bins in Spanish STS train

(b) Word share against relatedness bins in Spanish STS test

Figure 1.12: Word share against relatedness bins in Spanish STS train and Spanish STS test. *Word Share* indicates the ratio between number of common words in the two sentences to total number of words in the two sentences against each *Relatedness* bins

Sentence Pair	Similarity
1. Amás, los misioneros apunten que los númberos d'infectaos puen ser shasta dos o hasta cuatro veces más grandess que los oficiales. *(Furthermore, missionaries point out that the numbers of infected can be up to two or up to four times larger than the official ones.)* 2. Los cadáveres de personas fallecidas pueden ser hasta diez veces más contagiosos que los infectados vivos. *(The corpses of deceased people can be up to ten times more contagious than those infected alive.)*	0.6
1. La policía abatió a un caníbal cuando devoraba a una mujer Matthew Williams, de 34 años, fue sorprendido en la madrugada mordiendo el rostro de una joven a la que había invitado a su hotel. *(Police killed a cannibal while devouring a woman Matthew Williams, 34, was caught early in the morning biting the face of a young woman he had invited to his hotel.)* 2. La policía de Gales del Sur mató a un caníbal cuando se estaba comiendo la cara de una mujer de 22 años en la habitación de un hotel. *(South Wales police killed a cannibal when he was eating the face of a 22-year-old woman in a hotel room.)*	2
1. Ollanta Humala se reúne mañana con el Papa Francisco. *(Ollanta Humala meets tomorrow with Pope Francis.)* 2. El Papa Francisco mantuvo hoy una audiencia privada con el presidente Ollanta Humala, en el Vaticano. *(Pope Francis held a private audience today with President Ollanta Humala, at the Vatican.)*	3

Table 1.9: Example sentence pairs from the Spanish STS dataset. **Sentence Pair** column shows the two sentences. We also included their translations in the table. The translations were done by a native Spanish speaker. **Similarity** column indicates the annotated similarity of the two sentences.

is very different from the training set. As can be seen in Figure 1.10 training set has been annotated with relatedness scores 0-4 while the test set has been annotated with relatedness scores 0-5. Therefore, STS methods

Measure	Spanish STS Train		Spanish STS Test	
	Sent_1	**Sent_2**	**Sent_1**	**Sent_2**
Word Count Mean	31.23	31.02	9.03	9.34
Word Count STD	12.15	12.37	3.66	3.74
Word Count MAX	90	90	22	24
Word Count MIN	5	1	3	3

Table 1.10: Word count stats in Spanish STS training and Spanish STS testing.*STD* indicates the standard deviation and the other acronyms indicate the common meaning

should be able to handle that properly. Furthermore, as shown in Figure 1.11 and in Table 1.10 sentence pairs in test set are shorter in word length than the sentence pairs in train set. Therefore, STS methods working on this dataset should be able to properly handle that too. This can be observed as a weakness in this dataset, but that the same time this property of the dataset can be exploited to measure the strength of a STS system as well.

2. **Arabic STS Dataset** [11] The Arabic STS dataset we selected was also used for the Arabic STS subtask in SemEval 2017 Task 1: Semantic Textual Similarity Multilingual and Cross-lingual Focused Evaluation [14]. Unlike Spanish, no data from previous SemEval competitions were available since this was the first time an Arabic STS task was organised in SemEval. More information about the extracted sentences will be shown in the Table 1.11.

   To prepare the annotated instances, a subset of the English STS 2017 dataset has been selected and human translated into Arabic. Sentences have been

[11] The Arabic STS dataset can be downloaded at http://alt.qcri.org/semeval2017/task1/index.php?id=data-and-tools

Dataset	Pairs	Source
Trial	23	Mixed STS 2016
MSRpar	510	newswire
MSRvid	368	videos
SMTeuroparl	203	WMT eval.

Table 1.11: Information about the datasets used to build the Arabic STS training set. **Dataset** column expresses the acronym used describe the dataset. **Pairs** is the number of sentence pairs in that particular dataset and **Source** shows the source of the sentence pairs.

translated independently from their pairs. Arabic translation has been provided by native Arabic speakers with strong English skills in Carnegie Mellon University in Qatar. Translators have been given an English sentence and its Arabic machine translation5 where they have performed post-editing to correct errors. STS labels have been then transferred to the translated pairs. Therefore, annotation guidelines and the template will be similar to the English STS 2017 dataset. 1103 sentence pairs were available for training and 250 sentence pairs were available in the test set. Table 1.12 shows few pairs of sentences with their similarity score. The machine learning models require to predict a value between 0-5 which reflects the similarity of a given Arabic sentence pair.

1.2.3 Datasets on Different Domains

In order to experiment how our STS methods can be adopted in to different domains we have used two datasets from different disciplines which we introduce in this section.

Sentence Pair	Similarity
1. أحدهم يقلي لحما. *Someone is frying meat.* 2. أحدهم يعزف البيانو. *Someone plays the piano.*	0.250
1. أمرأة تظيف المكونات في الإناء. *A woman cleaning ingredients in the bowl.* 2. إمرأة تكسر ثلاثة بيضات في الإناء. *A woman breaks three eggs in a bowl.*	1.750
1. طفلة تعزف القيثارة. *A Child is playing harp.* 2. رجل يعزف القيثارة . *A man plays the harp.*	2.250
1. المرأة تقطع البصل الأخضر. *The woman chops green onions.* 2. إمرأة تقشر بصلة. *A woman peeling an onion.*	3.250
1. الأيل قفز فوق السياج. *The deer jumped over the fence.* 2. أيل يقفز فوق سياج الإعصار. *Deer Jumps Over Hurricane Fence*	4.800

Table 1.12: Example question pairs from the Arabic STS dataset. **Sentence Pair** column shows the two sentences. We also included their translations in the table. The translations were done by a native Arabic speaker. **Similarity** column indicates the annotated similarity of the two sentences.

1. **Bio-medical STS Dataset: BIOSSES** [12] - BIOSSES is the first and only benchmark dataset for biomedical sentence similarity estimation. [23]. The dataset comprises 100 sentence pairs, in which each sentence has been

[12]Bio-medical STS Dataset: BIOSSES can be downloaded from https://tabilab.cmpe.boun.edu.tr/BIOSSES/DataSet.html

selected from the TAC (Text Analysis Conference) Biomedical Summarisation Track- training dataset containing articles from the biomedical domain [13]. The sentence pairs have been evaluated by five different human experts that judged their similarity and gave scores ranging from 0 (no relation) to 4 (equivalent). The score range was described based on the guidelines of SemEval 2012 Task 6 on STS [9]. Besides the annotation instructions, example sentences from the bio-medical literature have been also provided to the annotators for each of the similarity degrees. To represent the similarity between two sentences we took the average of the scores provided by the five human experts. Table 1.13 shows few examples in the dataset. The machine learning models require to predict a value between 0-4 which reflects the similarity of the given bio medical sentence pair.

A dataset as small as this one can not be used by to train a supervised ML method, requiring alternative approaches such as unsupervised methods and transfer learning techniques which we will be exploring in the next few chapters.

2. **Clinical STS Dataset: MedSTS** [14]. MedSTS is another important STS benchmark dataset built on electronic clinical records (EHR). MedSTS contains 1,642 sentence pairs which were employed in Track 1 of National NLP

[13]Biomedical Summarisation Track is a shared task organised in TAC 2014 - https://tac.nist.gov/2014/BiomedSumm/

[14]Clinical STS Dataset: MedSTS can be downloaded from https://n2c2.dbmi.hms.harvard.edu/track1.

Sentence Pair	Similarity
1. It has recently been shown that Craf is essential for Kras G12D-induced NSCLC. 2. It has recently become evident that Craf is essential for the onset of Kras-driven non-small cell lung cancer.	4
1. Up-regulation of miR-24 has been observed in a number of cancers, including OSCC. 2. In addition, miR-24 is one of the most abundant miRNAs in cervical cancer cells, and is reportedly up-regulated in solid stomach cancers.	3
1. These cells (herein termed TLM-HMECs) are immortal but do not proliferate in the absence of extracellular matrix (ECM) 2. HMECs expressing hTERT and SV40 LT (TLM-HMECs) were cultured in mammary epithelial growth medium (MEGM, Lonza)	1.4
1.The up-regulation of miR-146a was also detected in cervical cancer tissues. 2. Similarly to PLK1, Aurora-A activity is required for the enrichment or localisation of multiple centrosomal factors which have roles in maturation, including LATS2 and CDK5RAP2/Cnn.	0.2

Table 1.13: Example question pairs from the Arabic STS dataset. **Sentence Pair** column shows the two sentences. We also included their translations in the table. The translations were done by a native Arabic speaker. **Similarity** column indicates the averaged annotated similarity of the two sentences.

Clinical Challenges (n2c2): Clinical Semantic Textual Similarity. Out of the 1,642 sentence pairs, 1,068 pairs were from the BioCreative/OHNLP 2018 shared task [24] [15] as well as 1,006 new pairs from two EHR systems, GE [16]

[15]BioCreative/OHNLP shared task is available on https://sites.google.com/view/ohnlp2018/home

[16]GE Healthcare provides IT healthcare solutions that also includes EHR solutions and can be accessed from https://www.gehealthcare.sa/products/healthcare-it/electronic-medical-records

and Epic [17]. Sentence pairs for BioCreative/OHNLP 2018 shared task have been extracted from Mayo Clinic's clinical data warehouse [25].

The creators of the dataset have removed protected health information (PHI) in the sentences by employing a frequency filtering approach [26]. Once the sentence pairs have been selected two clinical experts have being asked to annotate each sentence pair on the basis of their semantic equivalence. The annotation guideline is similar to the annotation guidelines of the STS 2017 English dataset [9]. Table 1.14 and Table 1.15 shows some example sentence pairs from the dataset with the gold labels and their explanations. The machine learning models require to predict a value between 0-5 which reflects the similarity of the given sentence pair.

1.3 Evaluation Metrics

While training a model is a key step, how the model generalizes on unseen data is an equally important aspect that should be considered in every machine learning model. We need to know whether it actually works and, consequently, if we can trust its predictions. This is typically called as *evaluation*. All of the datasets that we introduced in the previous section has what we call a *test* set. The machine learning models need to provide their predictions for the test test and the predictions will be evaluated against the true values of the test set.

There are three common evaluation metrics that are employed in Semantic

[17] Epic is a cloud-based EHR solution. More information can be viewed from their website https://www.epic.com/

Sentence Pair	Relatedness
The two sentences are completely equivalent as they mean the same thing. 1. Albuterol [PROVENTIL/VENTOLIN] 90 mcg/Act HFA Aerosol 2 puffs by inhalation every 4 hours as needed. 2. Albuterol [PROVENTIL/VENTOLIN] 90 mcg/Act HFA Aerosol 1-2 puffs by inhalation every 4 hours as needed 1 each.	5
The two sentences are completely equivalent as they mean the same thing. 1. Discussed goals, risks, alternatives, advanced directives, and the necessity of other members of the surgical team participating in the procedure with the patient. 2. Discussed risks, goals, alternatives, advance directives, and the necessity of other members of the healthcare team participating in the procedure with the patient and his mother.	4
The two sentences are roughly equivalent, but some important information differs/missing. 1. Cardiovascular assessment findings include heart rate normal, Heart rhythm, atrial fibrillation with controlled ventricular response. 2. Cardiovascular assessment findings include heart rate, bradycardic, Heart rhythm, first degree AV Block.	3

Table 1.14: Example sentence pairs from the MedSTS dataset with their gold relatedness scores (on a 5-point rating scale) and explanations - Part I. **Sentence Pair** column shows the two sentence and **Relatedness** column denotes the annotated relatedness score.

Textual Similarity tasks, which we explain in this section. We will be using them to evaluate our models through out the first part of our research.

In the equations presented for each of the evaluation metrics, we represent the gold labels with X and predictions with Y. Therefore, a gold label in i^{th} posi-

Sentence Pair	Relatedness
The two sentences are not equivalent, but share some details. 1. Discussed risks, goals, alternatives, advance directives, and the necessity of other members of the healthcare team participating in the procedure with (patient) (legal representative and others present during the discussion). 2. We discussed the low likelihood that a blood transfusion would be required during the postoperative period and the necessity of other members of the surgical team participating in the procedure.	2
The two sentences are not equivalent, but are on the same topic. 1. No: typical 'cold' symptoms; fever present (greater than or equal to 100.4 F or 38 C) or suspected fever; rash; white patches on lips, tongue or mouth (other than throat); blisters in the mouth; swollen or 'bull' neck; hoarseness or lost voice or ear pain. 2. New wheezing or chest tightness, runny or blocked nose, or discharge down the back of the throat, hoarseness or lost voice.	1
The two sentences are completely dissimilar 1. The risks and benefits of the procedure were discussed, and the patient consented to this procedure. 2. The content of this note has been reproduced, signed by an authorized	0

Table 1.15: Example sentence pairs from the MedSTS dataset with their gold relatedness scores (on a 5-point rating scale) and explanations - Part II. **Sentence Pair** column shows the two sentence and **Relatedness** column denotes the annotated relatedness score.

tion will be represented by X_i and a prediction in i^{th} position will be represented by Y_i.

1. **Pearson's Correlation Coefficient** - Correlation is a technique for investigating the relationship between two quantitative, continuous variables. Pearson's correlation coefficient (ρ) is a measure of the strength of the linear association between the two variables. A value of +1 is total positive linear correlation between the variables, 0 is no linear correlation, and -1 is total negative linear correlation.

   Pearson's Correlation Coefficient is one of the most common evaluation metrics in STS shared tasks [7, 9, 15, 16, 17, 18]. A machine learning model with a Pearson's Correlation Coefficient close to 1 indicates that the predictions of that model and gold labels have a strong positive linear correlation and therefore, it is a good model to predict STS. Pearson's Correlation Coefficient equation is shown in Equation 1.1 where *cov* is the covariance, σ_X is the standard deviation of X and σ_Y is the standard deviation of Y.

$$\rho = \frac{\mathrm{cov}(X, Y)}{\sigma_X \sigma_Y} \tag{1.1}$$

2. **Spearman's Correlation Coefficient** - Spearman's Correlation Coefficient (τ) is another common evaluation metric in STS shared tasks [7, 9, 15, 16, 17, 18]. It assesses how well the relationship between two variables can be described using a monotonic function. A monotonic relationship is a relationship that does one of the following:

   (a) as the value of one variable increases, so does the value of the other variable, *OR*,

(b) as the value of one variable increases, the other variable value decreases.

But not exactly at a constant rate whereas in a linear relationship the rate of increase/decrease is constant. The fundamental difference between Pearson's Correlation Coefficient and Spearman's Correlation Coefficient is that the Pearson Correlation Coefficient only works with a linear relationship between the two variables whereas the Correlation Coefficient works with the monotonic relationships as well. Spearman's Correlation Coefficient equation is shown in Equation 1.2 where D_i is the pairwise distances of the ranks of the variables X_i and Y_i and n is the number of elements in X or Y.

$$\tau = 1 - \frac{6 \sum D_i^2}{n(n^2 - 1)} \tag{1.2}$$

3. **Root Mean Squared Error** - Both Pearson's Correlation Coefficient and Spearman's Correlation Coefficient works only when both gold labels(X) and predictions (Y) are continues. Therefore, in the datasets like Quora Question Pairs where the gold labels are discrete values, Root Mean Squared Error (RMSE) is preferred for evaluation than Correlation Coefficient values. RMSE measures the distance between the gold labels and the predictions. RMSE equation is shown in Equation 1.3 where n is the number of

elements in X or Y.

$$rmse = \sqrt{(\frac{1}{n})\sum_{i=1}^{n}(Y_i - X_i)^2} \tag{1.3}$$

1.4 Contributions

The main contributions of this part of the thesis are as follows.

1. In each chapter, we cover various techniques to compute semantic similarity at the sentence level that can benefit the applications in the machine translation domain.

2. We propose a novel unsupervised STS method that outperforms current state of the arts unsupervised STS methods in all the English datasets, non-English datasets and datasets in other domains.

3. We propose a novel Siamese neural network architecture model which is efficient and outperforms current Siamese neural network architectures in all STS datasets.

4. We provide important resources to the community. The code of the each chapter as an open-source GitHub repository and the pre-trained STS models will be freely available to the community. The link to the GitHub repository and the models will be unveiled in the introduction section of the each chapter.

Chapter 2

State of the Art Methods

The biggest challenge that the neural based architectures face when applied to STS tasks is the small size of datasets available to train them. As a result, in many cases the networks cannot be trained properly. Given the amount of human labour required to produce datasets for STS, it is not possible to have high quality large training datasets. As a result researches working in the field have also considered unsupervised methods for STS. Recent unsupervised approaches use pretrained word/sentence embeddings directly for the similarity task without training a neural network model on them. Such approaches have used cosine similarity on sent2vec [27], InferSent [28], Word Mover's Distance [29], Doc2Vec [30] and Smooth Inverse Frequency with GloVe vectors [31]. While these approaches have produced decent results in the final rankings of shared tasks, they have also provided strong baselines for the STS task.

This chapter explores the performance of three unsupervised STS methods - cosine similarity using average vectors, Word Mover's Distance [29] and cosine similarity using Smooth Inverse Frequency [31] and how to improve them using contextual word embeddings which will be explained more in Section 2.1. The main contributions of this part of the thesis are as follows.

1. In the Related Work Section (Section 2.1), we cover three unsupervised STS techniques to compute semantic similarity at the sentence level.

2. We propose an improved unsupervised STS method based on contextual word embeddings.

3. The code with the experiments conducted will be publicly available to the community[1].

4. We published the findings in this chapter in Ranasinghe et al. [32].

The rest of this chapter is organised as follows. Section 2.1 describes the three unsupervised STS methods we experimented in this section. In section 2.2 we present the methodology, the contextual word embeddings we used followed by the results to the English datasets comparing with the baselines. Section 2.3 and Section 2.4 shows how our method can be applied to different languages and domains and their results. The chapter finishes with conclusions and ideas for future research directions in unsupervised STS methods.

2.1 Related Work

Given that a good STS metric is required for a variety of natural language processing fields, researchers have proposed a large number of such metrics. Before the shift of interest in neural networks, most of the proposed methods relied

[1] The public GitHub repository is available on https://github.com/tharindudr/simple-sentence-similarity

heavily on feature engineering. With the introduction of word embedding models, researchers focused more on neural representation for this task. The three unsupervised STS methods explored in this paper: Cosine similarity on average vectors, Word Mover's Distance and Cosine similarity using Smooth Inverse Frequency are the most common unsupervised methods explored in STS tasks. Apart from them cosine similarity of the output from Infersent [28], sent2vec [27] and doc2vec [30] have been used to represent the similarity between two sentences which we discuss in the next chapter.

2.1.1 Cosine Similarity on Average Vectors

The first unsupervised STS method that we used to estimate the semantic similarity between a pair of sentences, takes the average of the word embeddings of all words in the two sentences, and calculates the cosine similarity between the resulting embeddings. This is a common way to acquire sentence embeddings from word embeddings. Obviously, this simple baseline leaves considerable room for variation. Researches have investigated the effects of ignoring stopwords and computing an average weighted by tf-idf in particular.

2.1.2 Word Mover's Distance

The second STS state of the arts method that we have considered is Word Mover's Distance introduced by Kusner et al. [29]. Word Mover's Distance uses the word embeddings of the words in two texts to measure the minimum distance that the words in one text need to "travel" in semantic space to reach the words in the other text as shown in Figure 2.1. Kusner et al. [29] says that this is a good approach than vector averaging since this technique keeps the word vectors as it is through out the operation.

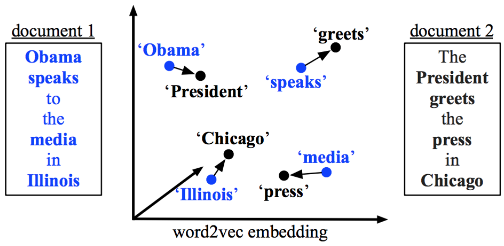


Figure 2.1: The Word Mover's Distance between two sentences

2.1.3 Cosine Similarity Using Smooth Inverse Frequency

The third and the last unsupervised STS method we have considered is to acquire sentence embeddings using Smooth Inverse Frequency proposed by Arora et al. [31] and then calculate the cosine similarity between those sentence embeddings. Semantically speaking, taking the average of the word embeddings in a sentence tends to give too much weight to words that are quite irrelevant. Smooth Inverse Frequency tries to solve this problem in two steps.

1. Weighting: Smooth Inverse Frequency takes the weighted average of the word embeddings in the sentence. Every word embedding is weighted by $\frac{a}{a+p(w)}$, where a is a parameter that is typically set to 0.001 and $p(w)$ is the estimated frequency of the word in a reference corpus.

2. Common component removal: After that, Smooth Inverse Frequency computes the principal component of the resulting embeddings for a set of

sentences. It then subtracts their projections on first principal component from these sentence embeddings. This should remove variation related to frequency and syntax that is less relevant semantically.

As a result, Smooth Inverse Frequency downgrades unimportant words such as *but, just*, etc., and keeps the information that contributes most to the semantics of the sentence. After acquiring the sentence embeddings for a pair of sentences, the cosine similarity between those two vectors were taken to represent the similarity between them.

All of these STS methods are based on word embeddings/vectors. The main weakness of word vectors is that each word has the same unique vector regardless of the context it appears. For an example, the word "play" has several meanings, but in standard word embeddings such as GloVe [33], FastText [34] or Word2Vec [35] each instance of the word has the same representation regardless of the meaning which is used. As an example the word 'bank' in two sentences - "I am walking by the river bank" and "I deposited money to the bank" would have the same embeddings which can be confusing for machine learning models. The recent introduction of contextualised word representations solved this problem by providing vectors for words considering their context too. In this way the word 'bank' in above sentences have two different embeddings. Therefore, we explore how the contextualised word representations can improve the above mentioned unsupervised STS methods. The following contextualised word representation models were considered for the experiments. We will explain the neural network architectures of these contextual word embeddings in Chapter 5.

For this chapter, we considered these architectures as a black box where we just feed the words to get the embeddings. We considered these contextualised word representations mainly considering the popularity they had by the time we were doing the experiments.

1. **ELMo**[2] introduced by Peters et al. [36] use bidirectional language model (biLM) to learn both word (e.g., syntax and semantics) and linguistic context. After pre-training, an internal state of vectors can be transferred to downstream natural language processing tasks. ELMo vectors have been successfully used many natural language processing tasks like text classification [37], named entity recognition [38] which motivated us to explore ELMo in unsupervised STS methods. Also, we were aware about the fact that ELMo has been pre-trained on different languages [39] and different domains [40] which will be easier when we are adopting our methodology for different languages and domains in Sections 2.3 and 2.4.

2. **BERT**[3] introduced by Devlin et al. [4] might probably be the most popular contextualised word embedding model. In contrast to ELMo which uses a shallow concatenation layer [4], BERT employs a deep concatenation layer. As a result BERT is considered a very powerful embedding architecture. BERT has been successfully applied in many natural language processing tasks like text classification [41], word similarity [42], named

[2]More details about ELMo can be viewed on `https://allennlp.org/elmo`
[3]The GitHub repository of BERT is available on `https://github.com/google-research/bert`

entity recognition [43], question and answering [44] etc. Similar to ELMo, BERT too has been widely adopted to different languages[4] such as Arabic [45], French [46], Spanish [47], Greek [48] etc. and different domains such as SciBERT [49], BioBERT [50], LEGAL-BERT [51] etc.

3. **Flair**[5] is another type of popular contextualised word embeddings introduced in Akbik et al. [52]. It takes a different approach by using a character level language model rather than the word level language model used in ELMo and BERT. Flair also has been used successfully in natural language processing tasks such as named entity recognition [53], part-of-speech tagging [52] and has been widely adopted in to different languages and domains [52, 54].

Apart from using these contextual word embedding models individually we also considered **Stacked Embeddings** of these models together. Stacked Embeddings are obtained by concatenating different embeddings. According to Akbik et al. [52] stacking the embeddings can provide powerful embeddings to represent words. Therefore, we experimented with several combinations of Stacked Embeddings.

Even though these contextual word embedding models have shown promising results in many natural language processing tasks, to the best of our knowledge none of these contextual word representations has been applied on unsupervised STS methods.

[4]Information about pretrained BERT models for different languages can be found on `https://bertlang.unibocconi.it/`

[5]The GitHub repository of Flair is available on `https://github.com/flairNLP/flair`

2.2 Improving State of the Art STS Methods

As mentioned before we applied different contextual word embeddings on three unsupervised STS methods and their variants. First we experimented with English STS datasets we explained in Section 1.2. Our implementation was based on *Flair-NLP* Framework [55] which makes it easier to switch between different word embedding models when acquiring word embeddings. Also *Flair-NLP* has their own model zoo of pre-trained models to allow researchers to use state-of-the-art NLP models in their applications. For English, all of these contextualised word embedding models come with different variants like *small, large etc.*. Usually the larger models provide a better accuracy since they have been trained on a bigger dataset compared to the smaller models. However, this comes with the disadvantage that these larger models are resource-intensive than the smaller models. In order to achieve a better accuracy, we used the largest model available in each contextual word embedding models. We will describe them in the following paragraphs.

For ELMo we used the 'original (5.5B)' pre-trained model provided in Peters et al. [36] which was trained on a dataset of 5.5B tokens consisting of Wikipedia (1.9B) and all of the monolingual news crawl data from WMT[6] 2008-2012 (3.6B). Peters et al. [36] mentions that ELMo original (5.5B) has slightly higher performance than other ELMo models and recommend it as a default model. Using this model we represented each word as a vector with a size of 3072 values.

[6]WMT: Workshop on Statistical Machine Translation is a leading conference in NLP that is being organised annually.

For BERT we used the 'bert-large-cased' pre-trained model. Compared to the 'bert-base-cased' model, this model provided slightly better results in all the NLP tasks experimented in Devlin et al. [4]. We represented each word as a 4096 lengthened vector using this model.

As suggested in Akbik et al. [52] the recommended way to use Flair embeddings is to stack pre-trained 'news-forward' flair embeddings and pre-trained flair 'news-backward' embeddings with GloVe [56] word embeddings. We used the stacked model to represent each word as a 4196 lengthened vector.

As mentioned before we also considered stacked embeddings of ELMo and BERT. For this we used pre-trained 'bert-large-uncased' model and 'original (5.5B)' pre-trained ELMo model to represent each word as a 4096 + 3072 vector.

In order to compare the results of contextualised word embeddings, we used a standard word representation model in each experiment as a baseline. In this research we used Word2vec embeddings [57] pre-trained on Google news corpus[7]. We represented each word as a 300 lengthened vector using this model.

In the following list we show the performance of each unsupervised STS method with contextual word embeddings on different English STS datasets.

1. **Cosine Similarity on Average Vectors** - The first unsupervised STS method we tried to improve using contextual word embeddings is Cosine Similarity on Average Vectors which we explained on Section 2.1. Table 2.1 shows the results for SICK dataset, Table 2.2 shows the results for STS

[7]Pretrained Word2vec can be downloaded from https://code.google.com/archive/p/word2vec/

2017 dataset. In order to compare our results with the other systems, we conducted the experiments only on the test data of the three mentioned datasets. Since this method leaves considerable room for variation, we have investigated the following variations and reported their results in each table.

(a) All the word vectors were considered for averaging. Results are shown in column I of Tables 2.1, 2.2 and 2.3

(b) All the word vectors except the vectors for stop words were considered for averaging. Column II of Tables 2.1, 2.2 and 2.3 shows the results.

(c) All the word vectors were weighted from its tf-idf scores and considered averaging. Results are shown in column III of Tables 2.1, 2.2 and 2.3

(d) Stop words were removed first and remaining word vectors were weighted from its tf-idf scores and considered averaging. Column IV of Tables 2.1, 2.2 and 2.3 shows the results.

From the results in Table 2.1, 2.2 and 2.3 there is no clear indication that contextualised word embeddings perform better than the standard word embeddings. In all the datasets considered, the best result was provided by Word2vec.

All the contextualised word embedding models we considered have more than 3000 dimensions for the word representation which is significantly

	I		II		III		IV	
Model	ρ	τ	ρ	τ	ρ	τ	ρ	τ
Word2vec	**0.730**†	0.624	**0.714**	0.583	**0.693**	0.570	**0.687**	0.555
ELMo	0.669	0.592	0.693	0.603	0.676	**0.579**	0.668	**0.572**
Flair	0.646	0.568	0.670	0.562	0.644	0.535	0.643	0.531
BERT	0.683	0.633	0.686	0.606	0.557	0.552	0.539	0.538
ELMo $\bigoplus$ *BERT*	0.696	**0.634**†	0.702	**0.614**	0.607	0.562	0.591	0.551

Table 2.1: Results for SICK dataset with Vector Averaging. I, II, III and IV indicates the different variations as explained before. For each word embedding model, Pearson Correlation (ρ) and Spearman Correlation (τ) are reported on all variations between the predicted values and the gold labels of the test set. $\bigoplus$ indicates a stacked word embedding model. Best result in each variation is marked in **Bold**. Best result from all the variations is marked with †.

	I		II		III		IV	
Model	ρ	τ	ρ	τ	ρ	τ	ρ	τ
Word2vec	**0.625**†	0.583	0.609	**0.635**	**0.640**	**0.591**	**0.588**	**0.573**
ELMo	0.575	0.574	**0.618**	0.609	0.374	0.395	0.352	0.376
Flair	0.411	0.444	0.584	0.586	0.325	0.374	0.336	0.386
BERT	0.575	0.574	0.555	0.588	0.355	0.401	0.309	0.386
ELMo $\bigoplus$ *BERT*	0.600	**0.597**†	0.591	0.608	0.391	0.413	0.354	0.398

Table 2.2: Results for STS 2017 dataset with Vector Averaging. I, II, III and IV indicates the different variations as explained before. For each word embedding model, Pearson Correlation (ρ) and Spearman Correlation (τ) are reported on all variations between the predicted values and the gold labels of the test set. $\bigoplus$ indicates a stacked word embedding model. Best result in each variation is marked in **Bold**. Best result from all the variations is marked with †.

higher than the number of dimensions for the word representation we had for standard embeddings - 300. As the vector averaging model is highly dependent on the number of dimensions that a vector can have, the curse of dimensionality might be the reason for the poor performance of contextualised word embeddings.

2. **Word Mover's Distance** - As the second unsupervised STS method we

	I	**II**	**III**	**IV**
Model	RMSE	RMSE	RMSE	RMSE
Word2vec	**0.621**	**0.591**†	**0.646**	**0.607**
ELMo	0.629	0.615	0.652	0.649
Flair	0.720	0.711	0.743	0.735
BERT	0.651	0.643	0.673	0.662
ELMo $\bigoplus$ *BERT*	0.625	0.611	0.650	0.647

Table 2.3: Results for QUORA dataset with Vector Averaging. I, II, III and IV indicates the different variations as explained before. For each word embedding model, Root Mean Squared Error (RMSE) is reported on all variations. $\bigoplus$ indicates a stacked word embedding model. Best result in each variation is marked in **Bold**. Best result from all the variations is marked with †.

experimented with Word Mover's Distance explained on Section 2.1. We have investigated the effects of considering/ ignoring stop words before calculating the word mover's distance which are detailed below.

(a) Considering all the words to calculate the Word Mover's Distance. Results are shown in column I of Tables 2.4, 2.5 and 2.6

(b) Removing stop words before calculating the Word Mover's Distance. Column II of Tables 2.4, 2.5 and 2.6 shows the results.

As depicted in table contextualised word representations could not improve Word Mover's method too over standard word representations. Since the travelling distance is dependent on number of dimensions, the curse of dimensionality might be the reason for the poor performance of contextualised word representations in this scenario too.

	I		II	
Model	ρ	τ	ρ	τ
Word2vec	**0.730**†	0.624	**0.714**	0.583
ELMo	0.669	0.592	0.693	0.603
Flair	0.646	0.568	0.670	0.562
BERT	0.683	0.633	0.686	0.606
ELMo $\bigoplus$ *BERT*	0.696	**0.634**†	0.702	**0.614**

Table 2.4: Results for SICK dataset with Word Mover's Distance. I and II indicates the different variations as explained before. For each word embedding model, Pearson Correlation (ρ) and Spearman Correlation (τ) are reported on all variations between the predicted values and the gold labels of the test set. $\bigoplus$ indicates a stacked word embedding model. Best result in each variation is marked in **Bold**. Best result from all the variations is marked with †.

	I		II	
Model	ρ	τ	ρ	τ
Word2vec	**0.625**†	0.583	0.609	**0.635**
ELMo	0.575	0.574	**0.618**	0.609
Flair	0.411	0.444	0.584	0.586
BERT	0.575	0.574	0.555	0.588
ELMo $\bigoplus$ *BERT*	0.600	**0.597**†	0.591	0.608

Table 2.5: Results for STS 2017 dataset with Word Mover's Distance. I and II indicate the different variations as explained before. For each word embedding model, Pearson Correlation (ρ) and Spearman Correlation (τ) are reported on all variations between the predicted values and the gold labels of the test set. $\bigoplus$ indicates a stacked word embedding model. Best result in each variation is marked in **Bold**. Best result from all the variations is marked with †.

2.3 Portability to Other Languages

2.4 Portability to Other Domains

2.5 Conclusions

	I	II
Model	RMSE	RMSE
Word2vec	**0.621**	**0.591**†
ELMo	0.629	0.615
Flair	0.720	0.711
BERT	0.651	0.643
ELMo $\bigoplus$ *BERT*	0.625	0.611

Table 2.6: Results for QUORA dataset with Word Mover's Distance. I and II indicate the different variations as explained before. For each word embedding model, Root Mean Squared Error (RMSE) is reported on all variations. $\bigoplus$ indicates a stacked word embedding model. Best result in each variation is marked in **Bold**. Best result from all the variations is marked with †.

Chapter 3

Sentence Encoders

3.1 Introduction

[28]

3.2 Related Work

3.3 Exploring Sentence Encoders in English STS

3.4 Portability to Other Languages

3.5 Portability to Other Domains

3.6 Conclusions

Chapter 4

Siamese Neural Networks

4.1 Introduction

[58] *Siamese Neural Networks*

4.2 Related Work

4.3 MAGRU: Improving Siamese Neural Networks

4.3.1 Portability to Other Languages

4.3.2 Portability to Other Domains

4.4 Conclusions

CHAPTER 5

TRANSFORMERS

5.1 Introduction

[4]

5.2 Related Work

5.3 Exploring Transformers in English STS

5.4 Exploring Transformers for STS in Other Languages

5.5 Exploring Transformers for STS in Other Domains

5.6 Conclusions

Part II

Applications - Translation Memories

Chapter 1

Introduction

1.1 What is Translation Memory?

[59]

1.2 Datasets

1.3 Related Work

1.4 STS for Translation Memories

Chapter 2

Sentence Encoders for Translation Memories

2.1 Introduction

[60]

2.2 Methodology

2.3 Results and Evaluation

Chapter 3

Future of Translation Memories

3.1 Introduction

3.2 Is Deep learning is the future for TMs?

3.3 Future Directions

Part III

Applications - Translation Quality Estimation

Chapter 1

Introduction

1.1 What is Translation Quality Estimation?

1.2 Datasets

1.3 Related Work

[61]

1.4 STS for Translation Quality Estimation

1.5 Conclusion

Chapter 2

TransQuest: STS Architectures for QE

2.1 Introduction

[62]

2.2 Methodology

2.3 Results and Evaluation

2.4 Conclusion

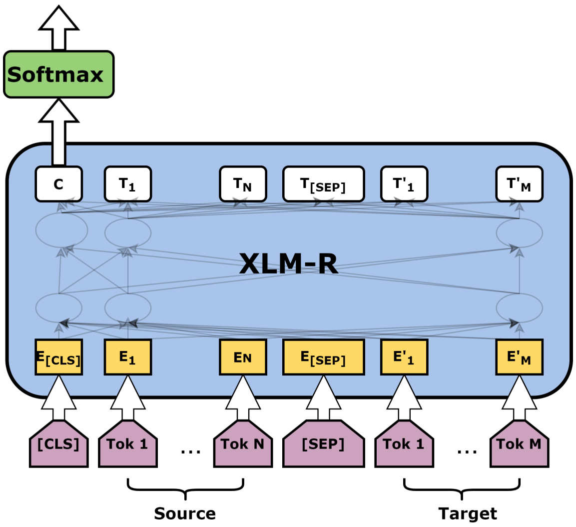


Figure 2.1: *MonoTransQuest* architecture

	Method	Mid-resource				High-resource			
		En-Cs SMT	En-Ru NMT	En-Lv SMT	En-Lv NMT	De-En SMT	En-Zh NMT	En-De SMT	En-De NMT
I	MTransQuest	**0.7207**	**0.7126**	0.6592	0.7394	**0.7939**	0.6119	0.7137	**0.5994**
	STransQuest	0.6853	0.6723	0.6320	0.7183	0.7524	0.5821	0.6992	0.5875
II	MTransQuest *-En|En-*	0.7168	0.7046	**0.7181**	**0.7482**	0.7939	0.6101	0.7355	0.5992
	STransQuest *-En|En-*	0.6663	0.6701	0.6533	0.7192	0.7524	0.5721	0.7000	0.5793
III	MTransQuest-m	0.7111	0.7012	0.7141	0.7450	0.7878	0.6092	0.7300	0.5982
	STransQuest-m	0.6561	0.6614	0.6621	0.7202	0.7369	0.5612	0.7015	0.5771
IV	Quest ++	0.3943	0.2601	0.3528	0.4435	0.3323	NR	0.3653	NR
	OpenKiwi	NR	0.5923	NR	NR	NR	0.5058	0.7108	0.4001
	Best system	0.6918	0.5923	0.6188	0.6819	0.7888	**0.6641**	**0.7397**	0.5718
V	mBERT	0.6423	0.6354	0.5772	0.6531	0.7005	0.5483	0.6239	0.5002

Table 2.1: Pearson (r) correlation between *TransQuest* algorithm predictions and human post-editing effort. Best results for each language by any method are marked in bold. Rows I, II and III indicate the different evaluation settings. Row IV shows the results of the state-of-the-art methods and the best system submitted for the language pair in that competition. **NR** implies that a particular result was *not reported* by the organisers. Row V presents the results of the multilingual BERT (mBERT) model in MonoTransQuest Architecture.

Chapter 3

Multilingual Quality Estimation with TransQuest

3.1 Introduction

[63]

3.2 Methodology

3.3 Results and Evaluation

3.4 Conclusion

Chapter 4

Extending TransQuest for word-level QE

4.1 Introduction

4.2 Related Work

[64]

4.3 Methodology

4.4 Results and Evaluation

4.5 Conclusion

		IT				Pharmaceutical			Wiki	
	Train Language(s)	En-Cs SMT	En-De NMT	En-De SMT	En-Ru NMT	De-En SMT	En-LV NMT	En-Lv SMT	En-De NMT	En-Zh NMT
I	En-Cs SMT	0.6081	(-0.09)	(-0.07)	(-0.09)	(-0.15)	(-0.02)	(-0.01)	(-0.10)	(-0.11)
	En-De NMT	(-0.17)	0.4421	(-0.06)	(-0.02)	(-0.18)	(-0.01)	(-0.02)	(-0.01)	(-0.08)
	En-De SMT	(-0.01)	(-0.05)	0.6348	(-0.67)	(-0.14)	(-0.06)	(-0.04)	(-0.06)	(-0.09)
	En-Ru NMT	(-0.14)	(-0.08)	(-0.16)	**0.5592**	(-0.12)	(-0.01)	(-0.03)	(-0.09)	(-0.08)
	De-En SMT	(-0.43)	(-0.23)	(-0.33)	(-0.31)	**0.6485**	(-0.29)	(-0.32)	(-0.25)	(-0.28)
	En-LV NMT	(-0.12)	(-0.09)	(-0.14)	(-0.03)	(-0.12)	0.5868	(-0.01)	(0.09)	(-0.08)
	En-Lv SMT	(-0.04)	(-0.16)	(-0.10)	(-0.09)	(-0.16)	(-0.01)	0.5939	(-0.15)	(-0.14)
	En-De NMT	(-0.11)	(-0.01)	(-0.08)	(-0.02)	(-0.14)	(-0.02)	(-0.04)	0.6013	(-0.06)
	En-Zh NMT	(-0.19)	(-0.08)	(-0.17)	(-0.03)	(-0.16)	(-0.03)	(-0.06)	(-0.07)	0.6402
II	All	**0.6112**	**0.4523**	**0.6583**	0.5558	0.6221	**0.5991**	**0.5980**	0.6101	0.6229
	All-1	(-0.01)	(-0.01)	(-0.05)	(-0.02)	(-0.12)	(-0.01)	(-0.01)	(-0.01)	(-0.05)
III	Domain	0.6095	0.4467	0.6421	0.5560	0.6331	0.5892	0.5951	0.6021	0.6210
IV	SMT/NMT	0.6092	0.4461	0.6410	0.5421	0.6320	0.5885	0.5934	0.6010	0.6205
V	Baseline-Marmot	0.4449	0.1812	0.3630	NR	0.4373	0.4208	0.3445	NR	NR
	Baseline-OpenKiwi	NR	NR	NR	0.2412	NR	NR	NR	0.4111	0.5583
	Best system	0.4449	0.4361	0.6246	0.4780	0.6012	0.4293	0.3618	**0.6186**	**0.6415**

Table 4.1: Target F1-Multi between the algorithm predictions and human annotations. Best results for each language by any method are marked in bold. Sections I, II and III indicate the different evaluation settings. Section IV shows the results of the state-of-the-art methods and the best system submitted for the language pair in that competition. **NR** implies that a particular result was *not reported* by the organisers. Zero-shot results are colored in grey and it shows the difference between the best result in that section for that language pair and itself.

Chapter 5

TransQuest++: Multi-Task Transformers for QE

5.1 Introduction

[65]

5.2 Methodology

5.3 Results and Evaluation

5.4 Conclusion

Bibliography

[1] Hanna Béchara, Hernani Costa, Shiva Taslimipoor, Rohit Gupta, Constantin Orasan, Gloria Corpas Pastor, and Ruslan Mitkov. MiniExperts: An SVM approach for measuring semantic textual similarity. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 96–101, Denver, Colorado, June 2015. Association for Computational Linguistics. doi: 10.18653/v1/S15-2017. URL https://www.aclweb.org/anthology/S15-2017.

[2] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1150. URL https://www.aclweb.org/anthology/P15-1150.

[3] Yang Shao. HCTI at SemEval-2017 task 1: Use convolutional neural network to evaluate semantic textual similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 130–133, Vancouver, Canada, August 2017. Association for Computational Lin-

guistics. doi: 10.18653/v1/S17-2016. URL https://www.aclweb.org/anthology/S17-2016.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

[5] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763, 2019.

[6] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[7] Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. SemEval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th International*

Workshop on Semantic Evaluation (SemEval 2014), pages 1–8, Dublin, Ireland, August 2014. Association for Computational Linguistics. doi: 10.3115/v1/S14-2001. URL https://www.aclweb.org/anthology/S14-2001.

[8] Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier. Collecting image annotations using Amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 139–147, Los Angeles, June 2010. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W10-0721.

[9] Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/S12-1051.

[10] Darnes Vilariño, David Pinto, Saúl León, Mireya Tovar, and Beatriz Beltrán. BUAP: Evaluating features for multilingual and cross-level semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 149–153, Dublin, Ireland, August

2014. Association for Computational Linguistics. doi: 10.3115/v1/S14-2022. URL https://www.aclweb.org/anthology/S14-2022.

[11] Rohit Gupta, Hanna Béchara, Ismail El Maarouf, and Constantin Orăsan. UoW: NLP techniques developed at the University of Wolverhampton for semantic similarity and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 785–789, Dublin, Ireland, August 2014. Association for Computational Linguistics. doi: 10.3115/v1/S14-2139. URL https://www.aclweb.org/anthology/S14-2139.

[12] André Lynum, Partha Pakray, Björn Gambäck, and Sergio Jimenez. NTNU: Measuring semantic similarity with sublexical feature representations and soft cardinality. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 448–453, Dublin, Ireland, August 2014. Association for Computational Linguistics. doi: 10.3115/v1/S14-2078. URL https://www.aclweb.org/anthology/S14-2078.

[13] Alexander Chávez, Héctor Dávila, Yoan Gutiérrez, Antonio Fernández-Orquín, Andrés Montoyo, and Rafael Muñoz. UMCC_DLSI_SemSim: Multilingual system for measuring semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 716–721, Dublin, Ireland, August 2014. Association for Computational Linguistics. doi: 10.3115/v1/S14-2128. URL https://www.aclweb.org/anthology/S14-2128.

[14] Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/S17-2001. URL https://www.aclweb.org/anthology/S17-2001.

[15] Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/S13-1004.

[16] Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. SemEval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland, August 2014. Association for Computational Linguistics. doi: 10.3115/v1/S14-2010. URL https://www.aclweb.org/anthology/S14-2010.

[17] Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor

Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263, Denver, Colorado, June 2015. Association for Computational Linguistics. doi: 10.18653/v1/S15-2045. URL https://www.aclweb.org/anthology/S15-2045.

[18] Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/S16-1081. URL https://www.aclweb.org/anthology/S16-1081.

[19] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://www.aclweb.org/anthology/D15-1075.

[20] Yuxia Wang, Fei Liu, Karin Verspoor, and Timothy Baldwin. Evaluating the Utility of Model Configurations and Data Augmentation on Clinical Se-

mantic Textual Similarity. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, pages 105–111, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.bionlp-1.11. URL https://www.aclweb.org/anthology/2020.bionlp-1.11.

[21] Junyi Li, Xuejie Zhang, and Xiaobing Zhou. ALBERT-Based Self-Ensemble Model With Semisupervised Learning and Data Augmentation for Clinical Semantic Textual Similarity Calculation: Algorithm Validation Study. *JMIR Med Inform*, 9(1):e23086, Jan 2021. ISSN 2291-9694. doi: 10.2196/23086. URL http://medinform.jmir.org/2021/1/e23086/.

[22] Z. Imtiaz, M. Umer, M. Ahmad, S. Ullah, G. S. Choi, and A. Mehmood. Duplicate questions pair detection using siamese malstm. *IEEE Access*, 8:21932–21942, 2020. doi: 10.1109/ACCESS.2020.2969041.

[23] Gizem Soğancıoğlu, Hakime Öztürk, and Arzucan Özgür. BIOSSES: a semantic sentence similarity estimation system for the biomedical domain. *Bioinformatics*, 33(14):i49–i58, 07 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx238. URL https://doi.org/10.1093/bioinformatics/btx238.

[24] Majid Rastegar-Mojarad, Sijia Liu, Yanshan Wang, Naveed Afzal, Liwei Wang, Feichen Shen, Sunyang Fu, and Hongfang Liu. Biocreative/ohnlp challenge 2018. In *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, BCB '18,

page 575, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450357944. doi: 10.1145/3233547.3233672. URL https://doi.org/10.1145/3233547.3233672.

[25] Stephen T Wu, Hongfang Liu, Dingcheng Li, Cui Tao, Mark A Musen, Christopher G Chute, and Nigam H Shah. Unified Medical Language System term occurrences in clinical notes: a large-scale corpus analysis. *Journal of the American Medical Informatics Association*, 19(e1):e149–e156, 04 2012. ISSN 1067-5027. doi: 10.1136/amiajnl-2011-000744. URL https://doi.org/10.1136/amiajnl-2011-000744.

[26] Yanshan Wang, Naveed Afzal, Sunyang Fu, Liwei Wang, Feichen Shen, Majid Rastegar-Mojarad, and Hongfang Liu. Medsts: a resource for clinical semantic textual similarity. *Language Resources and Evaluation*, 54(1): 57–72, Mar 2020. ISSN 1574-0218. doi: 10.1007/s10579-018-9431-1. URL https://doi.org/10.1007/s10579-018-9431-1.

[27] Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. Unsupervised learning of sentence embeddings using compositional n-gram features. 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 528–540, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1049. URL https://www.aclweb.org/anthology/N18-1049.

[28] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D17-1070.

[29] Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 957–966. JMLR.org, 2015.

[30] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, page II–1188–II–1196. JMLR.org, 2014.

[31] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=SyK00v5xx.

[32] Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. Enhancing unsupervised sentence similarity methods with deep contextualised

word representations. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 994–1003, Varna, Bulgaria, September 2019. INCOMA Ltd. doi: 10.26615/978-954-452-056-4_115. URL https://www.aclweb.org/anthology/R19-1115.

[33] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

[34] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL https://www.aclweb.org/anthology/L18-1008.

[35] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA, 2013. Curran Associates Inc.

[36] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher

Clark, Kenton Lee, and Luke Zettlemoyer. 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, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://www.aclweb.org/anthology/N18-1202.

[37] Ye Jiang, Johann Petrak, Xingyi Song, Kalina Bontcheva, and Diana Maynard. Team bertha von suttner at SemEval-2019 task 4: Hyperpartisan news detection using ELMo sentence representation convolutional network. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 840–844, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/S19-2146. URL https://www.aclweb.org/anthology/S19-2146.

[38] Ling Luo, Nan Li, Shuaichi Li, Zhihao Yang, and Hongfei Lin. DUTIR at the CCKS-2018 Task1: A neural network ensemble approach for Chinese clinical named entity recognition. In *CEUR Workshop Proceedings*, volume 2242, pages 7–12. CEUR-WS, 2018.

[39] Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64,

Brussels, Belgium, October 2018. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/K18-2005.

[40] Qiao Jin, Bhuwan Dhingra, William Cohen, and Xinghua Lu. Probing biomedical embeddings from language models. In *Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP*, pages 82–89, 2019.

[41] Tharindu Ranasinghe, Marcos Zampieri, and Hansi Hettiarachchi. BRUMS at HASOC 2019: Deep learning models for multilingual hate speech and offensive language identification. In *CEUR Workshop Proceedings*, volume 2517, pages 199–207, 2019. URL http://ceur-ws.org/Vol-2517/T3-3.pdf.

[42] Hansi Hettiarachchi and Tharindu Ranasinghe. Brums at semeval-2020 task 3: Contextualised embeddings for predicting the (graded) effect of context in word similarity. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain, 2020. Association for Computational Linguistics.

[43] Chen Liang, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang. Bond: Bert-assisted open-domain named entity recognition with distant supervision. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '20, page 1054–1064, New York, NY, USA, 2020. Association for Comput-

ing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3403149. URL https://doi.org/10.1145/3394486.3403149.

[44] Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. End-to-end open-domain question answering with BERTserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-4013. URL https://www.aclweb.org/anthology/N19-4013.

[45] Wissam Antoun, Fady Baly, and Hazem Hajj. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France, May 2020. European Language Resource Association. ISBN 979-10-95546-51-1. URL https://www.aclweb.org/anthology/2020.osact-1.2.

[46] Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.645. URL https://www.aclweb.org/anthology/2020.acl-main.645.

[47] José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. Spanish Pre-Trained BERT Model and Evaluation Data. In *PML4DC at ICLR 2020*, 2020.

[48] John Koutsikakis, Ilias Chalkidis, Prodromos Malakasiotis, and Ion Androutsopoulos. GREEK-BERT: The Greeks Visiting Sesame Street. In *11th Hellenic Conference on Artificial Intelligence*, SETN 2020, page 110–117, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450388788. doi: 10.1145/3411408.3411440. URL `https://doi.org/10.1145/3411408.3411440`.

[49] Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1371. URL `https://www.aclweb.org/anthology/D19-1371`.

[50] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 09 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz682. URL `https://doi.org/10.1093/bioinformatics/btz682`.

[51] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.261. URL https://www.aclweb.org/anthology/2020.findings-emnlp.261.

[52] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1139.

[53] Alan Akbik, Tanja Bergmann, and Roland Vollgraf. Pooled contextualized embeddings for named entity recognition. 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 724–728, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1078. URL https://www.aclweb.org/anthology/N19-1078.

[54] Shreyas Sharma and Ron Daniel Jr. Bioflair: Pretrained pooled contextualized embeddings for biomedical sequence labeling tasks. *arXiv preprint arXiv:1908.05760*, 2019.

[55] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-4010. URL https://www.aclweb.org/anthology/N19-4010.

[56] Jeffrey Pennington, Richard Socher, and Christopher Manning. 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, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://www.aclweb.org/anthology/D14-1162.

[57] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013. URL http://arxiv.org/abs/1301.3781.

[58] Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. Semantic textual similarity with Siamese neural networks. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1004–1011, Varna, Bulgaria, September 2019. INCOMA

Ltd. doi: 10.26615/978-954-452-056-4_116. URL https://www.aclweb.org/anthology/R19-1116.

[59] Peter J. Arthern. Machine translation and computerized terminology systems: A translator's viewpoint. *Translating and the Computer, Proceedings of a Seminar, London 14th November 1978. Amsterdam: North-Holland Publishing Company*, pages 77–108, 1979.

[60] Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. Intelligent translation memory matching and retrieval with sentence encoders. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 175–184, Lisboa, Portugal, November 2020. European Association for Machine Translation. URL https://www.aclweb.org/anthology/2020.eamt-1.19.

[61] Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. OpenKiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 117–122, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-3020. URL https://www.aclweb.org/anthology/P19-3020.

[62] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual represen-

tation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.747. URL https://www.aclweb.org/anthology/2020.acl-main.747.

[63] Shuo Sun, Marina Fomicheva, Frédéric Blain, Vishrav Chaudhary, Ahmed El-Kishky, Adithya Renduchintala, Francisco Guzmán, and Lucia Specia. An exploratory study on multilingual quality estimation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 366–377, Suzhou, China, December 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.aacl-main.39.

[64] Varvara Logacheva, Chris Hokamp, and Lucia Specia. MARMOT: A toolkit for translation quality estimation at the word level. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3671–3674, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL https://www.aclweb.org/anthology/L16-1582.

[65] Dongjun Lee. Two-phase cross-lingual language model fine-tuning for machine translation quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1024–1028, Online, November 2020. Asso-

ciation for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.wmt-1.118.