Dissertation Type: Natural Language Processing

University of BRISTOL

DEPARTMENT OF COMPUTER SCIENCE

Summarizing Stance Dependent Claims on Twitter

Contextualising Opinions

William Nafack

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Bachelor of Science in the Faculty of Engineering.

Friday 21st May, 2021

Declaration

This dissertation is submitted to the University of Bristol in accordance with the requirements of the degree of BSc in the Faculty of Engineering. It has not been submitted for any other degree or diploma of any examining body. Except where specifically acknowledged, it is all the work of the Author.

William Nafack, Friday 21st May, 2021

Contents

Abstract

In this paper, I look at extracting the claims that summarise the opinions of tweets posted by users about a specific target. These opinions are classified into two groups, opinions supporting the target and opinions opposing the target. I look at developing a supervised framework consisting of three components: argument classification, stance detection, and extractive summarization. Although consistent research has been done in individual areas, there is little to no work that aims at unifying the components into one to generate these summaries.

Therefore the research hypothesis can be described as follows. Can combining these three components into a single system assist in producing informative summaries which effectively describe the general opinions supporting or contrasting a given target. To this regards, the set of objectives are stated as follows

- Experimenting with different word representation vectors and metadata features to develop an argument classification and stance detection model.
- Produce gold standard summaries to evaluate the summaries generated by the model.
- Implementation of an algorithm to perform extractive summarization based on clustering.
- Implementation of a complete system that integrates the argument classifier, stance classifier, and extractive summariser.
- Evaluate the results against the gold standard summaries.

Contributions

Below is stated the following contributions the paper has made.

- The paper demonstrated that simple models like the Bag of Word model with (1-3) n-grams and hashtag features perform better than a BERT model pre-trained on controversial topics for open-domain argument Reimers et al. 2019 on social media data like Twitter.
- The paper has shown that meta data features like hashtags plays an important role in classification task on social media data.
- The paper has shown that Linear SVC with BERT embeddings outperforms the baseline set by the task organizers in Mohammad et al. 2017 on tweets with opinions directed towards a target.
- The paper demonstrated that Clustering algorithms may not be fit for summarizing opinions of users on social media.

Achievements

The following states the achievement throughout the project.

- Implemented an argument classifier and a stance detector.
- Implemented an algorithm for extracting meaningful tweets based on K-medoids clustering with cosine distance.
- combined the three components into a single system.
- Created gold standard summaries for each target and stance label to evaluate the model's generated summary.
- Evaluated each component described in the paper.

Acknowledgements

I would like to thank my supervisor for his advice and guidance during this project as well as the support provided by my friends and family during this academic year.

Chapter 1

Introduction

Social Media has played a vital role in connecting people online and exchanging information. Hubs like Twitter have provided a platform on which users can share their thoughts and ideas in an instant via tweets. The continuous growth has led to a large amount of available information for others to interact with, making Twitter a center for topic discussions. Some are arguably controversial in nature. Such discussions can portray a selection of opinions in favor or against the target of the discussion. With millions of tweets sent every second, It suddenly becomes tough to navigate through this pile of information to find the claims that would best generalize the reasons supporting or contrasting a target.

Breaking down this problem would involve three aspects. That of argument classification to select the claims made towards a target, stance detection to differentiate between the claims supporting or contrasting the target, and extractive summarization to select the most meaningful claims. Although there is extensive research is in each of these areas, Interestingly enough, there has been little work when it comes to exploiting the combination of these three components into a framework to generate the claims that summarize the opinions of a particular stance on Twitter data. The only work found was that of Jang and Allan 2018 which focused on an Unsupervised classification of the user stance followed by a ranking model to rank the tweets to generate the extractive summaries—each component to be dealt with also come with their associated challenges. Argument classification on social media based on evidence is notoriously tricky as users mainly use it to connect with others and share information. Hence claims may not be as constructively written and present proof to back up the claim. Stance Detection on tweets is another rather complex issue. While stance detection on articles and legal documents shows excellent results, It would be completely different on tweets as many do not contain the target on which the stance is assessed. Extractive summarization work has also focused on articles and documents to get the key sentences. These methods are adapted to work on long sequences of text and can be less efficient when it comes to short text like tweets.

This project aims to develop a model that extracts the most salient claims supporting or contrasting a specific target. In this regard, I exploit the 3 component framework as a supervised task on the SemEval data set Mohammad et al. 2016a whose annotations suit our purposes.

One of the main challenges when dealing with social media text is its informal format. They consist of a short text of 280 characters and do not follow any writing style, rule, and metadata that may or may not add value to the tweet's context. As a result, It is challenging to determine the quality of an opinion. Although several works have been done in other domains to rank arguments qualitatively, such as that from Gretz et al. 2019, It is open for debate if the same can be applied to Twitter data. Furthermore, Tweets may or may not contain the target. This poses a significant challenge when attempting open domain target identification restricting the work on target specific evaluation. Finally, although we can evaluate each component separately, there is no known pipeline that would allow the evaluation of the whole framework making it complex to assess the margin error between the independent evaluation of sub system and the whole system.

With this in mind the set of objectives are described as follows.

- Experimenting with different word representation vectors and meta data features to develop an argument classification and stance detection model.
- Produce gold standard summaries to evaluate the summaries generated by the model.
- Implementation of an algorithm to perform extractive summarisation based on clustering.
- Implementation of a complete system that integrates the argument classifier, stance classifier and extractive summariser.
- Evaluate the results against the gold standard summaries.

Chapter 2

Technical Background

This chapter presents an overview of the related work to the three main components involved in the project and presents the theory behind the main distributional and distributed representations for text data in natural language processing used in the project.

2.1 Argument Classification

There has been several definition given to an Argument. According to Toulmin 2003, an argument consist of six components; a claim, data,qualifier ,warrant,rebuttal and backing.Early work by Moens et al. 2007 has used this reasoning to mine arguments from large corpus of legal papers and news articles.g In Amgoud et al. 2011, argument mining for reviews was introduced with the aim of extracting the reasons for positive or negative opinions.

In general, an argument is understood to consist of a claim and at least one evidence unit.In the case of social media and particularly Twitter,occurrences of such arguments may not be common as users are not only limited by the number of characters they can type(280 characters max) but also by the type of discussions they partake in.While all conversations are not argumentative in nature,those that are controversial can lead to arguments and opinions between users.For the purpose of this work,I treat a single claim or opinion as an argumentative unit with or without evidence similarly to the work of Bosc et al. 2016 and Addawood and Bashir 2016. However, the quality of an argumentative unit remains open for debate with different approaches taken such as the work of Gretz et al. 2019 who fine-tunes a BERT model on topic specific arguments.

Recent work on argument classification includes that of Reimers et al. 2019 who aims at open domain target argument classification and clustering by using a fine-tuned BERT(Bidirectional Encoder Representation from Transformers) Devlin et al. 2019 and Bidirectional Long-short term Memory(BiLSTMs).The experimental set up for the first task is targeted at topic dependent sentence-level argument classification in which they make use of of ELMo (Embeddings from Language Model) Peters et al. 2018 and BERT.The process involves a constructive fine tuning of the ELMo layers and BERT embeddings which can be deemed computationally expensive.In addition to this, the data set used is that of UKP Corpus from Stab et al. 2018 which contains 25,492 sentences on 8 controversial topics gotten from 400 documents.One question that needs to be asked however, is how well would such a system perform on Twitter data.Taking into account the complexity that comes with fine tuning the deep learning architectures on may argue it is perhaps easier to look at more simpler approaches that would rely on the contextual embeddings of the tweets and possible meta data features.

2.2 Stance Detection

According to BIBER and FINEGAN 1989, stance is defined as the expression of the speaker's standpoint and judgment toward a given proposition .It plays an imminent role in studies measuring public opinion.The nature of these issues is usually controversial as people express their opposing opinions particularly on political and social issues.

Early work in Stance detection includes that of Somasundaran and Wiebe 2010 which focus on capturing ideological stances in debates using opinion analysis. Thomas et al. 2006 research was aimed at analysing transcripts from the the U.S Congressional floor debates to understand whether the speeches support or opposed the proposed legislation.

Due to the globalization of internet and social media, the application of detecting stances has shifted toward different domains.In the case of social media like Twitter,It has gained a lot of significant research.One of those is the SemEval 2016 task for detecting stance in tweets created by Mohammad et al. 2016b.This focused on content-dependent features learned from labeled data for a closed set of topics.The data set produced as a result Mohammad et al. 2016a is annotated for stances, sentiment , target and opinions expressed making it useful for training stance detection models. Other researches which aim at identifying stances without prior open-domain target identification or topic knowledge include that of Darwish et al. 2019 which uses an unsupervised framework via a combination of metadata features such as retweeted accounts, UMAP for dimensionality reduction, and Mean Shift for clustering to detect the stance of a user on a controversial topic.

Another interesting research is the IBM's project debator on which part of its work was focused on detecting stance in claims from Wikipedia articles Bar-Haim et al. 2017.Their approach relies on sentiment analysis.This is interesting as there is a noticeable misconception between sentiment and stance.

Stance detection mainly focuses on identifying a person's standpoint or view toward an object of evaluation.It can take a different approach by leveraging non-textual features such as social meta data and contextual features to infer a user's stance. On the other hand sentiment analysis is typically approached as a linguistic agnostic task which mainly focus on these linguistic properties to identify the polarity of a text Benamara et al. 2017.Considering this may vary depending on the domain to which it is applied, in the case of Twitter data, Mohammad et al. 2017 has shown through a common classification task system for stance and sentiments that sentiment features are not as effective for stance detection as they are for sentiment analysis.

2.3 Extractive Summarization

Extractive summarization aims at identifying the salient information from a corpus that is then taken and grouped together to form a concise summary.Early work in this domain includes that of Nenkova and Vanderwende 2005 which weigh words according to their probability distribution in the corpus and utilize the most frequent occurring words.It is not be assimilated with Tf-idf (Term Frequency–Inverse Document Frequency) in which it considers both the frequency of the term in a document and inverse document frequency that is pages of document in which the required terms exist. Graph based methods such Nenkova and Vanderwende 2005 and Mihalcea and Tarau 2004 in the input document is represented as a connected graph. The vertices represent the sentences, and the edges between vertices have attached weights that show the similarity of the two sentences. The score of a sentence is the importance of its corresponding vertex, which can be computed using graph algorithms.

In the recent years,the approach has shifted towards the use of transformers and neural networks.Zhou et al. 2018 proposed a model which consist of a document encoder and sentence extractor build on top of a recurrent neural network and citation fine tuned a BERT model for extracting sentences.Others (Liu et al. 2019a treated the problem using dependency tree structures with nodes representing summary sentences and the use of pre-trained BERT embeddings Liu and Lapata 2019.

In the domain of social media data,Chavan and Suryawanshi 2016 focus on segmentation techniques to capture the semantic significance of tweets followed by clustering while Xu et al. 2013 approach the issue as a ranking problem on a graph based model.Latest work include that of Naik et al. 2018 which build on the segmentation techniques and uses a combination of Particle swarm Algorithm and Clustering.Another one is that of Chakraborty et al. 2019 for tweets related to news articles, which aim at solving issues relevance, diversity and coverage of the tweets using a community detection technique and greedy approximation of the minimum dominating set (MDS) based approach to select suitable tweets.

In the context of a complete framework that aims at the summarization of the opinions ,to the best of our knowledge only Jang and Allan 2018 focus on controversial topics and aimed at capturing the most salient tweets that would explain why the topic or target is controversial. This was done through an unsupervised framework to detect the stances of the users followed by a ranking model to select the summary tweets. A serious weakness with their methodology was that of the features used when training their linear regression model which evaluates the articulation of a tweet. This feature was a measure of offensive language and was based on a dictionary text file of "bad" words. This is not only very subjective and leads to bias in the model but also not a viable approach as the dictionary only contains a limited amount of words leading to skewed approximations.

2.4 Text Representation

Firth 1957 hypothesizes that words that occur in similar contexts have similar meanings.According to the distributional hypothesis, there is a high similarity between the meaning for words such as man and woman for example.As a results, if two words can occur in similar context , then the vectors representing these words must be close to each other.For example, 'fire' and 'dog' are two words which are not related in their meaning, may not be used in the same sentence. On the other hand, the words 'cat' and 'dog' are sometimes seen together, so they may share some aspect of meaning hence making their vector representation close to each other.

2.5 Distributional Representations

This involves the distribution of words from the context in which the words appear in a text Ferrone and Zanzotto 2020.The vectors formed can be considered as co-occurrence matrix that captures the co-occurence of words.Here the dimension of this matrix is equal to that of the size of the vocabulary of the corpus.

2.5.1 Bag of Words and Bag of N-Grams (BoW and BoN)

In BoW, the words are represented by their frequency as a binary or non-binary event in the text while ignoring the order and context. The intuition here is that a text that would belong to a specific class would be characterized by a unique set of words.Similarly, BoN follows a same representation with the only difference being in the words segmentation.BoN gives the possibility to represent a sentence as unigrams(one word sequence) , bigrams(sequence of two words), trigrams(sequence of three words) and more.The advantage over BoW is that it tries to incorporate a sense of ordering in the text by not exclusively treating words as independent units.

2.5.2 Term Frequency Inverse Document Frequency(TF-IDF)

Term frequency measures how often a term or word occurs in a given text.As a term can occur more in longer text than shorter text , it is normalised by calculating the total number of occurences of the term in the text by the length of the text.On the other hand Inverse document frequency aims at capturing the importance of a term across the whole corpus.By doing this , IDF weighs down the terms that are very common across the corpus and weighs up the unique terms.This gives the ability to then capture those words that may have a great signification in a text.Equation 2.1 and 2.2 shows their calculations respectively.

$$\text{Term Frequency}(t, d) = \frac{\text{Number of occurrences of term } t \text{ in document } d}{\text{Total Number of terms in the document } d} \tag{2.1}$$

$$\text{IDF}(t) = \log_e \frac{\text{Total number of documents in the corpus}}{\text{Number of documents with term } t \text{ in them}} \tag{2.2}$$

2.6 Distributed Representations

Basic vectorization approaches such as one-hot encoding , bag of words(BoW and BoN) and TF-IDF has brought about a number of limitations.

- They cannot handle Out of Vocabulary Words(OOV).This is a situation in which the data is transformed on a piece of text including a fix amount of words.As a result it does not recognize new words that can be found during training on a new corpus.
- one-hot encoding , BOW and tf-idf can results in very sparse and highly dimensional vectors.This can affect the learning capability of the machine especially if dense features are also used at the same time making the features represented in the sparse matrix computationally inefficient.
- They form discrete representations,and as such hamper the ability to capture relationship between words in which sequences and combinations of multiple words can provide valuable information

To this effect, the distributed representation concept by Ferrone and Zanzotto 2020 was developed. Distributed representation aims at compressing the highly dimensional and sparse vectors into low dimensional and dense vectors.

2.6.1 Word2Vec

Word2Vec is a model presented by Mikolov et al. 2013 presented a model trained on millions of text pieces to form salient vector representations of the words in its vocabulary.Conceptually,it takes a large corpus of text as input and ”learns” to represent the words in a common vector space based on the contexts in which they appear in the corpus.The objective is to have words with similar context occupy close spatial positions. Mathematically, the cosine of the angle between such vectors of similar context should be close to 1, that is the angle should be close to 0.This can be given by the cosine similarity.Given two vectors A and B their similarity is calculated as show in 2.3.

$$\text{similarity} = \cos\theta = \frac{A.B}{||A||_2||B||_2} \tag{2.3}$$

Word2Vec construct its word embeddings using Neural Networks with two different architectures , CBOW(Common Bag Of Words) and Skip Gram of which we used the gensim pre-trained model with CBOW architecture

Continuous Bag of Words (CBOW) In the CBOW architecture, A Language model is build to take the context in which a word is present as input and predict the ”center” word corresponding to that context.A Language model is a model that gives a probability distribution over sequences of words.More specifically, by one hot encoding the input word , the measure of the output error is compared to the one hot encoding target word. In the process of predicting the target word, we learn the vector representation of the target word.Skip-gram is the reverse as it learn to predict the context of words given the target.The vectors are the weights of a logistic regression model for making the predictions.2.1 shows the architecture of a simple CBOW model.

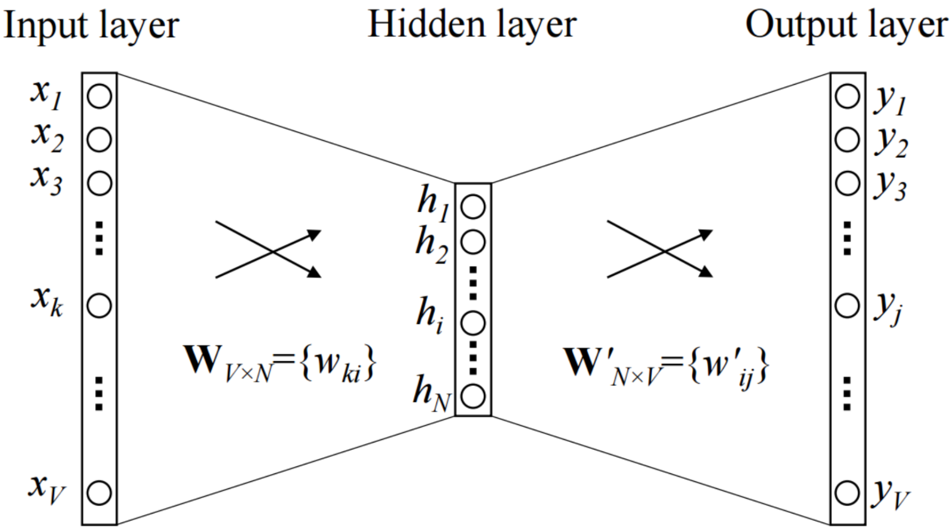
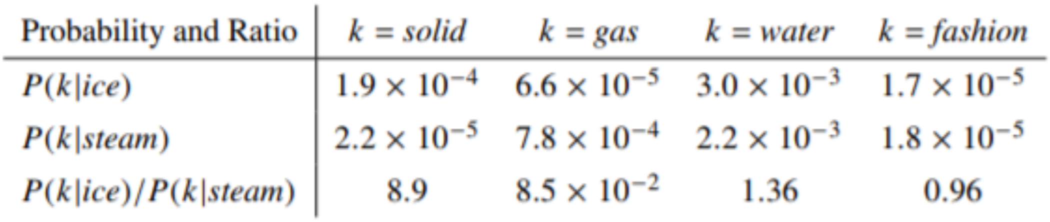


Figure 2.1: A simple CBOW model with only one word in the context

The input is a one hot encoded vector of size V. The hidden layer contains N neurons and the output is aslo of size V length vector with the elements being the softmax values.$W_{V\times N}$ is the weight matrix that maps the input x to the hidden layer (V, N) dimensional matrix.$W\prime_{N\times V}$ is the weight matrix that maps the hidden layer outputs to the final output layer (N, V) dimensional matrix.

2.6.2 GloVe

Glove is a vector space representation model developed by Pennington et al. 2014.Unlike Word2vec which relies only on local information of language; that is the semantics learnt for a given word, is only affected by the surrounding words,GloVe captures both global statistics and local statistics of a corpus, in order to come up with word vectors.Glove uses an unsupervised framework via Linear Discriminant Analysis(LDA) to produce low dimensional vectors by singular value decomposition on the co-occurrence matrix of words.A good point about Word2Vec is that similar words are located together in the vector space and arithmetic operations on word the vectors can pose semantic or syntactic relationships. However, LDA cannot maintain such linear relationship in a vector space.The motivation of GloVe is to force the model to learn such linear relationship based on the co-occurrence matrix explicitly.The ratios of word to word co-occurrence probabilities have the potential for encoding some form of meaning as can be seen from 2.2.



Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

Figure 2.2: Occurrence probabilities for target words ice and steam from Pennington et al. 2014

2.6.3 Sentence Embeddings :Bidirectional Encoder Representations from Transformers(BERT)

Representations we have seen so far gives each word a fixed representation.This can be an issue to some extent when dealing with words employed in different context.For example the sentences ”I have an ant bite on my arm” and ”It is important to arm yourself with a solid education” both uses the word ”arm”.However, their context is different in each of these sentence.The word embeddings approach seen so far can not find a direct way to capture this information.

BERT Devlin et al. 2019 has allowed the production of state of the art models in many natural language processing task like sentiment analysis, topic modelling , named entity recognition.The key idea is to leverage ”transfer learning”; that is to learn the embeddings on a common task with a large corpus and fine-tune the learning on a task-specific data. It makes use of a transformer, an attention mechanism that learns contextual relations between words in a sentence. In its vanilla form(no added customization) , a transformer includes two separate mechanisms.An encoder that reads the text input and a decoder that produces a prediction for the task. As BERT's goal is to generate a language model, only the encoder mechanism is necessary.The transformer is considered non-directional and hence allows the model to learn the context of word in relation to its neighbours from the left and right.BERT is pre-trained on two tasks namely Masked Language Modeling and Next Sentence Prediction.Before training the sentences are preprocessed using a specific set of rules as shown in 2.3.

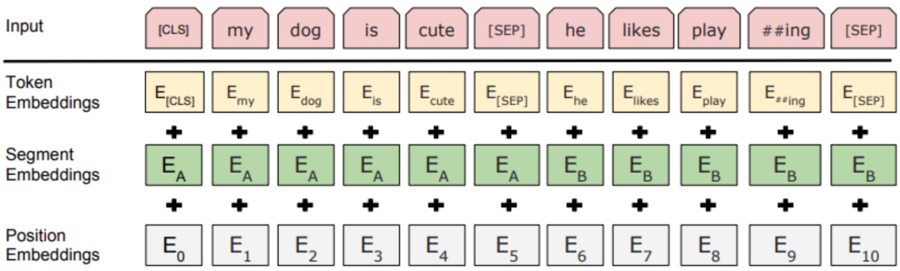

Input
[CLS]
my
dog
is
cute
[SEP]
he
likes
play
##ing
[SEP]
Token Embeddings
$E_{[CLS]}$ E_{my} E_{dog} E_{is} E_{cute} $E_{[SEP]}$ E_{he} E_{likes} E_{play} $E_{\#\#ing}$ $E_{[SEP]}$
Segment Embeddings
E_A E_A E_A E_A E_A E_A E_B E_B E_B E_B E_B
Position Embeddings
E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10}

Figure 2.3: Preprocessing of tokens in BERT architecture Devlin et al. 2019.

- A [CLS] token is inserted at the beginning of the first sentence and a [SEP] token is inserted at the end of each sentence.
- A sentence embedding indicating Sentence A or Sentence B is added to each token.
- A positional embedding is added to each token to indicate its position in the sequence Vaswani et al. 2017.

Masked Language Modeling (MLM)

Before inputting word tokens into BERT, 15% of the words in each sequence are replaced with a [MASK] token 2.4. The model will then proceed to predict the original value of the masked words, based on the context provided by the other, non-masked, words in the sequence.More specifically, the prediction of the words require the following.

- Adding a classification layer on top of the encoder output.
- Multiplying the output vectors by the embedding matrix, transforming them into the vocabulary dimension.
- Calculating the probability of each word in the vocabulary with softmax.

BERT loss function proceeds to look only the prediction of the masked words and ignores the prediction of the non-masked words.

Next Sentence Prediction

During the training process, the model receives pairs of sentences as input and aims at predicting if the second sentence in the pair is the subsequent sentence in the original corpus. During training, 50% of the inputs are pairs in which the second sentence is the subsequent sentence in the original corpus, while in the other 50% a random sentences from the corpus is chosen as the second sentence. The assumption is that the random sentence will be disconnected from the first sentence.To predict if the second sentence is truly connected to the first, the steps below are completed.

- The entire input sequence goes through the Transformer model.
- The output of the [CLS] token is transformed into a 2×1 shaped vector, using a simple classification layer.
- The probability of the next sequence is then calculated with softmax.

Both tasks are trained together so as to minimize the combined loss function of the two procedures.2.4 shows a high-level description of the transformer encoder. The input is a sequence of tokens, which are first embedded into vectors and then processed in the neural network. The output is a sequence of vectors of size H, in which each vector corresponds to an input token with the same index.

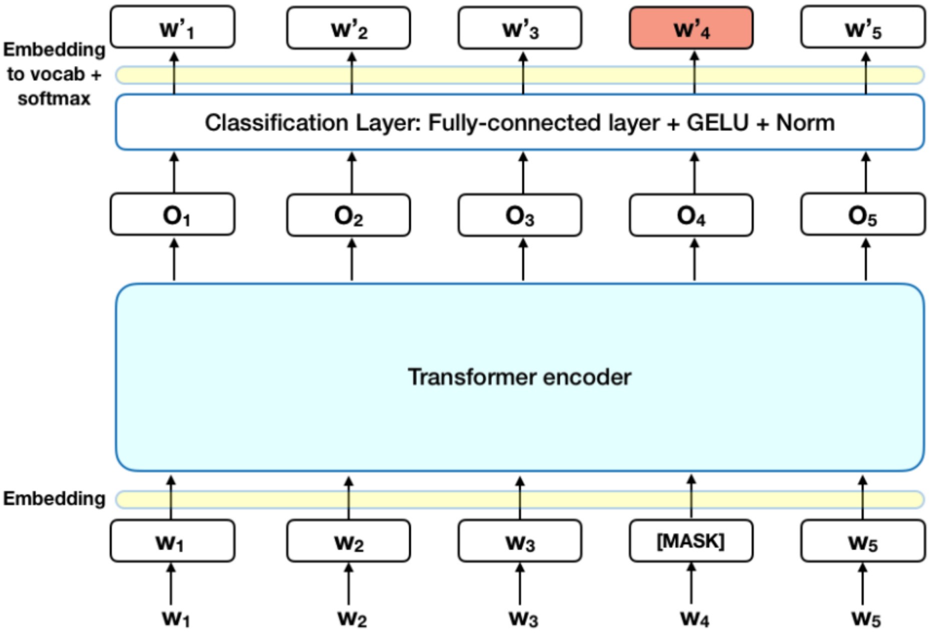

W'1
W'2
W'3
W'4
W'5
Embedding to vocab + softmax
Classification Layer: Fully-connected layer + GELU + Norm
O1
O2
O3
O4
O5
Transformer encoder
Embedding
W1
W2
W3
[MASK]
W5
W1
W2
W3
W4
W5

Figure 2.4: High level architecture of BERT Devlin et al. 2019

Extracting Sentence Representation

In order to generate sentence embeddings Reimers and Gurevych 2019 uses the output of the BERT model and adds a pooling operation to derive fixed sized sentence embeddings.The general idea introduced is to pass two sentences through BERT, in a Siamese network as shown in 2.5.

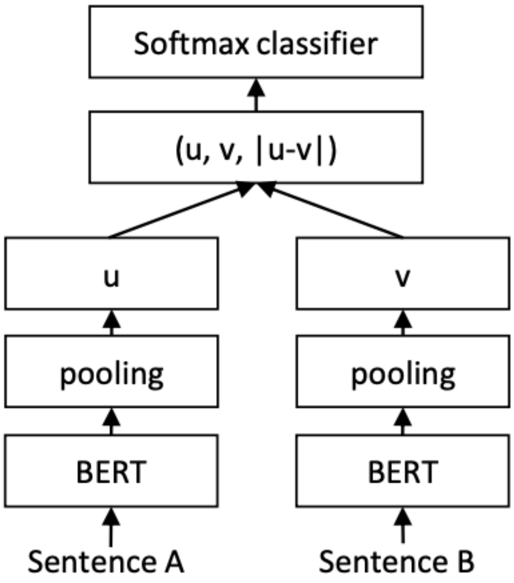


Figure 2.5: Pooling 2 Sentences from BERT via a Siamese network Reimers and Gurevych 2019.

The next step then involves the concatenation of the embeddings $(u, v, u - v)$, multiplied by a trainable weight matrix.The pooling strategy is flexible, although the authors found that a mean aggregation worked best.

Chapter 3

Supporting Technologies

The chapter outlines the technologies that were used, including a description of why various software was appropriate for the implementation.

- Python is the only language used throughout the whole development of the project. It was chosen as a programming language because of its list of packages that makes it suitable for creating machine learning models.
- Sklearn Pedregosa et al. 2011 was the main machine learning library used. It provides access to feature extraction , model selection ,classification and clustering methods to support the development of the models.
- Key packages such as Numpy Harris et al. 2020, Scipy Virtanen et al. 2020, Pandas McKinney et al. 2010 and matplotlib Hunter 2007 gives access to scientific computing and a large number of high level methods and functions which makes facilitate the development of models, analysis of data and insightful visualisation.
- Pytorch Paszke et al. 2019 is a deep learning library which I utilized to collapse the sentence embedding from BERT into vectors.
- nltk Loper and Bird 2002 is a multi purpose nlp library.The Tweet tokenization and stop-word removal methods from it were used during the preprocessing step.
- spacy Honnibal et al. 2020 is a nlp library which was used to produce summaries of the tweet via the textrank method and also produced the part of speech tags.
- Rouge Pltrdy is a python library which is a re-implementation of the Rouge metric for evaluation summaries.This was used to evaluate the Extractive summarization component.
- Gensim Rehurek and Sojka 2011 is a nlp library used to generate the pretrained word embeddings precisely word2vec and gloVe.
- Flair Akbik et al. 2019 is a nlp library used to generate pretrained sentence embeddings via transformers (BERT).
- Yellow brick Bengfort et al. 2018 is a visualisation tool which was used to visualise the Silhouette and Calinksi method when generating the optimal number of clusters.
- Wordcloud Oesper et al. 2011 is a python library used to generate wordcloud visualisation of text data. This visualisations show the words with highest frequency in a given corpus.
- I used the Sumy Miso-Belica package which has summarization methods to support my implementation of the SumBasic algorithm used a baseline in the extractive summarization component.
- Demoji bsolomon1124 and word ninja Keredson are python libraries used to translate emojis from text into their contextual meaning and to split concatenated words such as hashtags into separate words.

Chapter 4

Methodology

In this chapter, I present a discussion of the data set used, steps taken to prepare the data, and a detailed implementation of the theory presented in the technical background.

4.1 SemEval Stance Detection Data set

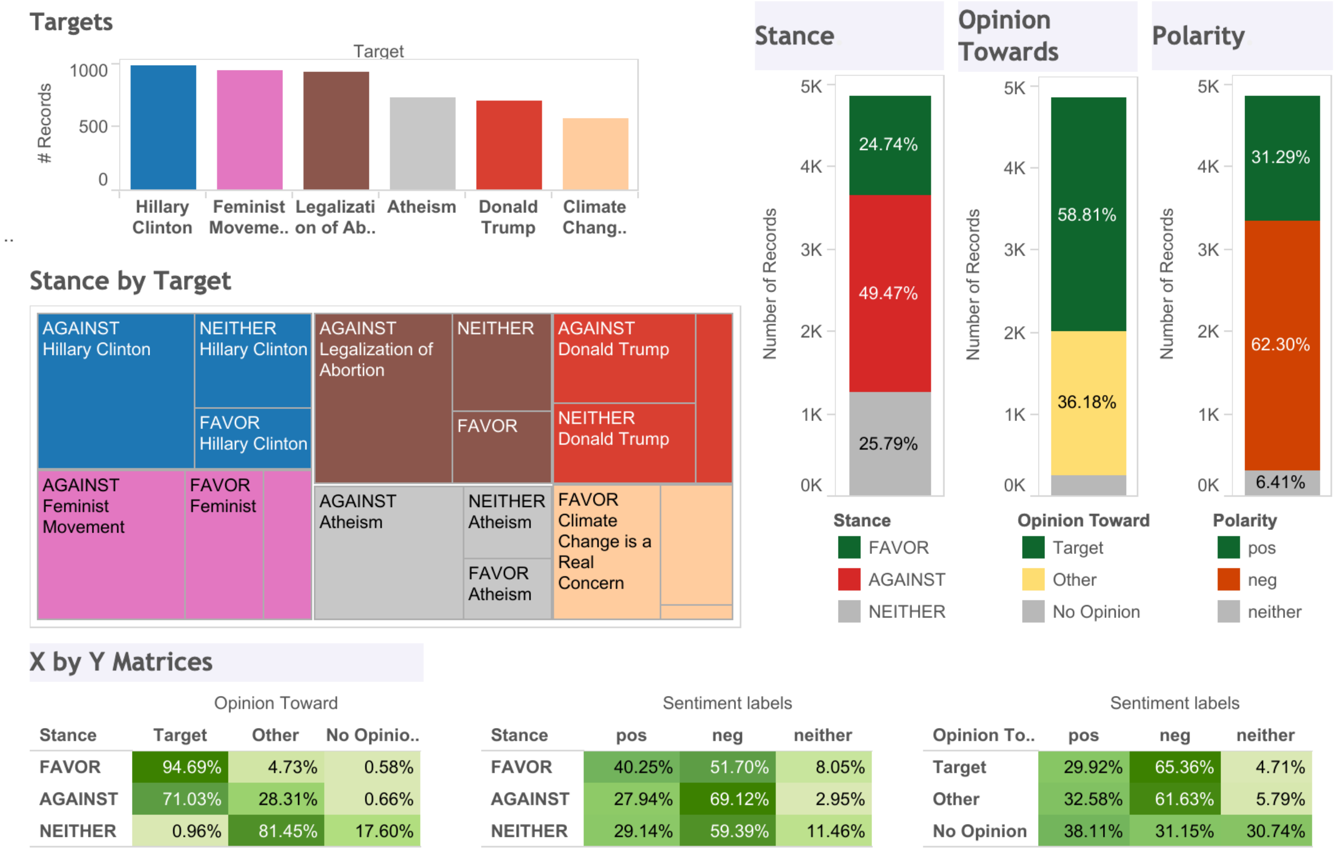
The SemEval Stance Dataset created by Mohammad et al. 2016a is annotated for stance, whether a tweet is in favor of or against a specific target of interest. The target of interest may or may not appear in the tweets or is not the target of opinion in the tweets. It is further annotated for sentiment, a binary annotation of whether a tweet's polarity is positive or negative. In addition to that, it is annotated for whether the target of interest is the target of opinion. This allows me to set a claim that was expressed to a target as argumentative and the inverse as not argumentative.

There are six targets in the data set, namely Abortion, Atheism, Climate Change, Real Concern, Feminist Movement, and Hillary Clinton. These targets have often been the center of debates on social media platforms. The authors provided training data in the form of 2,814 tweets covering the five targets with 395 to 664 tweets for a topic. The authors used crowd-sourcing to annotate these tweets manually. Class balance varied between topics, with some topics being significantly skewed. Climate Change is a Real Concern with only 4% *against* and 54% favor labels. while others are more balanced such as Feminist Movement with 49% *against* and 32% *favor* labels. Approximately 74% of the provided tweets were judged to be either in favor or against.Figure 4.1 shows the distribution of the different class label for all targets.

4.1.1 Annotated Summaries

The Data set is not annotated for summaries of the tweets for a specific target stance and opinion. To resolve this, I created gold standard summaries that would fit these intended purposes to evaluate my generated summaries properly. This summary should represent the claims presented towards a particular stance (in this case, I am only interested in those in favor and against) to a particular subject or topic. This would serve as a means for the user to understand the controversy along with this subject. More explicitly, opinions clearly stated in the context of a target that can explain what it is about that target that pushed the inclination to a favorable or opposing stance. The following criteria were taken into consideration when creating the gold standard references.

- Stance must be indicated in the Tweet:A gold standard tweet should reflect the stance of the user towards the target or topic.
- Target Relevance : A reference tweet is self-explanatory and relevant in the context of a particular topic or topic.tweets with no target makes it difficult to assess its context with regards to the target.



	Opinion Toward		
Stance	Target	Other	No Opinio..
FAVOR	94.69%	4.73%	0.58%
AGAINST	71.03%	28.31%	0.66%
NEITHER	0.96%	81.45%	17.60%

	Sentiment labels		
Stance	pos	neg	neither
FAVOR	40.25%	51.70%	8.05%
AGAINST	27.94%	69.12%	2.95%
NEITHER	29.14%	59.39%	11.46%

	Sentiment labels		
Opinion To..	pos	neg	neither
Target	29.92%	65.36%	4.71%
Other	32.58%	61.63%	5.79%
No Opinion	38.11%	31.15%	30.74%

Figure 4.1: Distribution of the data set taken from Mohammad et al. 2016a

- Articulation : A reference tweet avoids obscene language,it is clear and logical and can be understood by any user.

By studying the data set for each target this was the conclusion to the sort of tweets that were present.

- Argumentative (non-discrete): The tweet expresses a clear argument toward the subject or topic.
  Target : Hillary Clinton
  Tweet : Hillary is our best choice if we truly want to continue being a progressive nation. #Ohio

- Informative : The tweet informs about something which is related or holds in the context of the subject or topic.
  Target : Climate Change is a Real Concern
  Tweet: Around 1500 new homes being built every year in England in areas at high risk of #floods, adding to risk from #UKClimate2015 #SemST

- Narrative / Descriptive : The tweet states an experience or story from the user which is related or holds in the context of the subject or topic.
  Target : Feminist Movement
  Tweet : wanted to pursue a career rather than settle down and start a family in my 20's. Even now, people still tell me time is ticking #SemST

- Affirmative : The tweets agrees or supports with something which is related or holds in the context of the subject or topic
  Target :Hillary Clinton
  Tweet : I'm proud to announce I support #HillaryClinton!!!! #SemST"

- interrogative : The tweets present a question about something which is related or holds in the context of the subject or topic.

> *Target : Abortion*
> *Tweet: We all are sinners, but what right do humans have to kill another human being? Is that what God wanted us to do? #SCL #thoughts #SemST*

Each property of these tweets are not mutually exclusive hence a tweet could present one or more of these properties.I prioritised tweets with an argumentative aspect.

4.2 Data Preprocessing and Cleaning

Social media data presents a non-exhaustive list of challenges when it comes to preprocessing it in a suitable form for natural language processing task.Tweets essentially consist of 240 characters which can contains a mix of succinct information and other characteristics that have to be dealt with.This includes

- Meta data and special characters such as Hashtags , user mentions , third party links , emoticons, gifs, non-ASCII characters.
- Nonstandard spelling such as contracted forms of certain nouns and verbs and nonsensical punctuation.
- Multilingual in nature; presence of more than one language used in the text.

This can add noise to the text and hinder the process of capturing meaningful information from the words present in it. To account for this, the following steps are taken to effectively clean the data.The tweets are set to lowercase, Unicode and non-ASCII characters are removed. Any trailing white space and consecutive characters is dealt with.URLs and mentions are removed and hashtags are extracted as a feature as I consider it an important parameter to be experimented with.Furthermore, English stop words are removed as they can be found too often in text and do not add any additional information.I use regular expressions to remove nonsensical punctuation and symbols. I proceed with Tweet segmentation to break downs the tweet into tokens using Loper and Bird 2002 TweetTokenizer and this set of tokens are later used in our pipeline.I extract Part of speech tags with Honnibal et al. 2020 to identify the different part of speech present in the tweets.

4.3 Feature Engineering

The cleaned tweet are transformed into vector representations so as to feed them into the classification models.

All classification and summarization task involved use the same distributed representations.Using the Gensim Rehurek and Sojka 2011 Keyed Vectors method, I obtained the word2vec and gloVe embeddings from the clean tweets.The dimensionality of the word embeddings decide on the space of the learned embeddings.I use a dimension of 300 and 200 for word2vec and gloVe respectively. To produce the features vectors , rather than averaging embeddings for each individual word in the text, I decided to weigh each word embedding with its TF-IDF value calculated using the TF-IDF vectorizer from Scikit-learn Pedregosa et al. 2011.As a result, unique words in the vector space will have more weight to them hence increasing the ability to capture essential information.The result produce is a single vector with a dimension.

Subsequently, the sentence embeddings are produce using the flair library Akbik et al. 2019 which provides state of the art pre-trained BERT models.I used the RoBERTa(Robustly optimized BERT approach) Liu et al. 2019b and create the contextual vectors from the corpus.RoBERTa, is a retraining of BERT with improved training methodology, 1000% more data and compute power.To improve the training procedure, RoBERTa removes the Next Sentence Prediction (NSP) task from BERT's pre-training and introduces dynamic masking so that the masked token changes during the training epochs.RoBERTa has an edge on giving the best prediction metrics.

For Argument Classification and Stance detection the baseline features involved using Bag of Ngrams with (1-3) word n-grams and Bag of Ngrams with a combination of (1-3) and (2-5) word and character n-grams as used in the Mohammad et al. 2016b. Specifically for Argument Classification I experimented with different hybrid features to analyse their effects.This included

- The ratio of Part of speech tags present in each tweet.
- The number of hashtags in the tweet.
- The vector representation of the set of hashtags contained in a tweet using bag of words. The assumption here is that using this would help the model to separate noisy hashtags such as those used in all the tweets.For example #SemST to unique hashtags that can convey some information #HillaryForTheWin.

4.4 Classification Task

I experimented with several classifiers such as Logistic Regression, Support Vector Machines ,Artificial Neural Networks and Random Forest Regression using the scikit-learn Pedregosa et al. 2011 implementations of these models.I initially ran all the base models with no fine tuning and chose the one that showed the best results.

4.4.1 Hyperparamter Tuning and Validation

I performed cross validation over a constructed parameter space to find the best parameters. I make use of Grid Search from Scikit-learn with Stratified K-fold to obtain more representative samples as I am dealing with an imbalance data set.

4.5 Summarization Task

In this part of the task, the tweets are grouped according to each target in the corpus . I separated the tweets according to the two ground-truth stances (favour and against) across each target and chose only the tweets in which an opinion was annotated as expressed towards the target, a part of the target or an aspect of the target.

The processed tweets from this new corpus are then converted into their vector representations as stated in the feature engineering step.The extractive summarization technique is then applied onto each set of tweets such as to evaluate it individually.This is done using clustering algorithm to get the tweets that would contain the most information.I made use of Kmedoid algorithm Bezdek et al. 1984 and utilize the Calinsksi Kaoungku et al. 2018 and Silhouette method Kaoungku et al. 2018 to get the optimal number of clusters by taking the average of the two metrics using the scikit-learn implementations.Figure 4.2 and 4.3 shows hows plot of scores with varying number of clusters.

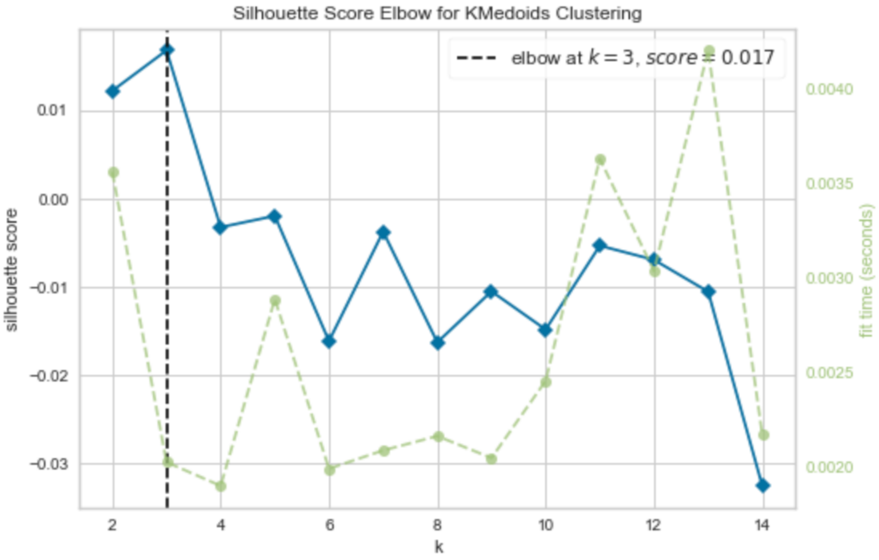


Figure 4.2: Silhouette score on Clustering with K-medoids by consine distance

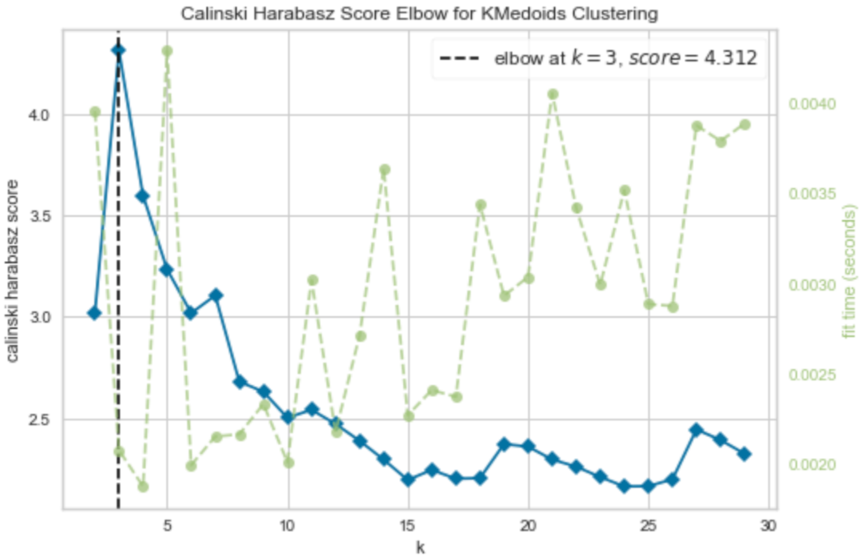


Figure 4.3: Calinski score on Clustering with k-medoids by cosine distance

The kmedoids algorithm is a partitioning clustering approach that separates the data set of n data points into K predefined distinctive non overlapping group called clusters where each data point is attributed to a cluster. k-medoids clustering is more stable and less susceptible to outliers and noise in comparison to k-means clustering algorithm. This is because it uses medoids , a single point as cluster centres instead of an average of points as used in K-means.The main point of using Kmedoids clustering is to diminish the summation of dissimilarities between data points in a given cluster and the respective cluster centre. The clusters are generated using the cosine similarity distance metric to measure the semantic similarity between words.The cost of K-medoid algorithm is shown in equation 4.1.

$$C = \sum_{P_i \epsilon C_i} \sum_{P_i \epsilon C_i} |P_i - C_i| \tag{4.1}$$

The pseudocode for the K-medoid algorithm is as follows.

- Select the medoids by radomly selecting k points from a set of n points.

- Each data point is clustered to its closest medoid using the cosine distance.
- While the cost reduces for every medoid C,for every data point p not selected as medoid:
    1. Exchange c and p , cluster each data point to the closest medoid, compute the cost again.
    2. Compare total cost with the ones in the previous step; If greater, do not assign data point as new medoid.

4.5.1 Selection of Tweets

After the clustering process, the tweets are selected from each clusters.The following pseudocode describes the selection method for k clusters generated and extraction of a summary of length N.

- initialise a list of sentences.
- set index of data point to zero
- set number of sentences to zero.
- initialise cluster count starting with first cluster.
- while the number of tweets is not equal to N.
    1. sort data points in terms of their cosine distances to the medoid in ascending order.
    2. extract the data point with minimum cosine distance to the medoid.
    3. add data point (sentence) to the list of sentences.
    4. increment cluster count by 1 modulo k.
    5. increment number of sentences.
    6. if number of sentences modulo k is 0, increment index of data point.

4.6 Three Component Combination

The predicted output from the argument classifier is used to classify the tweets as argumentative or non-argumentative. The tweets which are predicted as argumentative are then fed into the stance detection for each target. The tweets which support and oppose the target are then summarized using the k-medoids Clustering step. This gives the final summary generated from the data.

Chapter 5

Critical Evaluation

There is no system in place that allows evaluating end-to-end predictions of different models as it becomes complex to measure the noise attributed from each step. This is why the direction taken is to evaluate each component individually, as this would give a more general insight into the overall performance. In this chapter, I evaluate the performance of each component individually. The chapter begins by describing some of the metrics and baseline models used in evaluating the performance of the models. Subsequently, I proceed to show the results of each system, followed by some discussions. I terminate with a comparison of the models to alternative work in the same field for each component.

5.1 F1 Score

Accuracy is simply the ratio of the correct predictions against the total number of observations. Accuracy works well in situations where there is an equal number of observations per class, otherwise the accuracy score can be heavily skewed and not provide an understanding of the data which applies to the data set that I am experimenting with.Before discussing F1, it is important in understanding some key terms used in computing the F1 scores:

- True Positives : when the model predicted YES and the actual output was YES.
- True Negatives : when the model predicted NO and the actual output was NO.
- False Positives : when the model predicted YES and the actual output was NO.
- False Negatives : when the model predicted NO and the actual output was YES.

$$F1 = 2 \times \frac{1}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}} \tag{5.1}$$

The F1 Score, as seen in Equation 5.1, is simply the harmonic mean of precision and recall, where precision and recall are defined in Equations 5.2 and 5.3, respectively.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{5.2}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{5.3}$$

A high precision value but a low recall value signifies that the model is accurate but misses a significant number of observations that are considered difficult to classify. The main use of the F1 score is in informing how many observations are correctly classified, while also explaining how often the model misclassifies an observation. The aim is to maximize the F1 score.

5.2 Recall-Oriented Understudy for Gisting Evaluation(ROUGE)

ROUGE is a package for evaluating summaries developed by Lin 2004.It is a set of metrics but I mainly focus on two.Namely ROUGE-N and ROUGE-L.ROUGE-N measures the number of matching n-grams between the model generated text and a reference summary which is mainly human annotated.the 'N' represents the number of n-gram used. For ROUGE-1 , ROUGE-2, ROUGE-3 I measure the match-rate of unigrams,bigrams and trigrams between the model output and reference respectively.I measure the ROUGE F1 scores using the recall and precision values.

The recall counts the number of overlapping n-grams found in both the model's generated text and reference. It then divides this number by the total number of n-grams in the reference.

$$\text{Recall} = \frac{\text{Number of n-grams in model and reference}}{\text{number of n-grams in reference}} \tag{5.4}$$

It is great for ensuring the model is capturing all of the information contained in the reference.However, it is not good at ensuring my model is not just generating a huge number of words to increase the recall score.To avoid this I use the precision metric which is calculated in almost the exact same way, but rather than dividing by the reference n-grams count, It is divided by the model n-gram count.

$$\text{Precision} = \frac{\text{Number of n-grams in model and reference}}{\text{Number of n-grams in model}} \tag{5.5}$$

The F1 score is then generated using the same equation for calculating F1 scores as shown above.

ROUGE-L measures the longest common sub-sequence (LCS) between the model output and reference.To put it simply, It count the longest sequence of tokens that is shared between the model and reference.The idea here is that a longer shared sequence would indicate more similarity between the two sequences.I apply the recall and precision calculations just like before but this time replacing the match with LCS.

ROUGE is a great evaluation metric but comes with some drawbacks. In particular, ROUGE does not cater for different words that have the same meaning as it measures syntactical matches rather than semantics.Hence if I have two sequences that have the same meaning but uses different words to express that meaning they could be assigned a low ROUGE score.This can be offset slightly by using several references and taking the average score, but this will not solve the problem entirely.Nonetheless, it's a good metric for assessing both machine translation and automatic summarization tasks and is very popular for both.An alternative solution is that of Kann et al. 2018 (do we want to include this here or after the analysis to explain alternatives that combat limitations of the project)

5.3 SumBasic and TextRank

SumBasic Nenkova and Vanderwende 2005 and Textrank Mihalcea and Tarau 2004 are the two baselines used to evaluate the extractive summarization component.SumBasic uses a greedy search algorithm to score the sentences and hence aims at optimizing the problem at the expense of results.Evaluating my model against this allows me to compare not only the computational cost but how effective it is as compared to a somewhat randomized approach.Textrank essentially generates a cosine similarity matrix in which the similarity of one sentence to the other can be obtained.As the Clustering is performed using cosine similarity with the same intuition of having similar sentences clustered together , It makes it ideal to compare the effectiveness of the clustering.

5.3.1 SumBasic

SumBasic begins by simply counting the frequency of each token for individual sentences in the corpus. Each sentence is graded based on the probability of the tokens in it as a whole. The summary is then generated using a simple greedy search algorithm, which iteratively selects the sentence with the highest scoring tokens.This process is repeated until the maximum length of the summary is reached. SumBasic updates the probabilities of the tokens in the selected sentence by squaring them, modelling the likelihood of a word in order to avoid selecting the same or similar sentence multiple times.

5.3.2 TextRank

TextRank is a graph based method for extractive summarization inspired by the page rank algorithm.A graph is created from the corpus.The nodes represent the sentences, while the weights on the edges between two nodes is the similarity between them which can be found using cosine similarity.It follows an iterative process to find important weights from each node until consistent weights are found.The sentences are then sorted in a descending order based upon their scores choosing the first k sentences as summary.

5.4 Results

5.4.1 Classification Task

Both classification tasks made use of word and sentence embeddings to represent the text. Argument classification was evaluated on all targets using the F1 score metric. On the other hand, stance detection is tested on all targets and individual targets. The evaluation is done on the claims which contain opinion directed towards a target. I used the F1 score evaluation from Mohammad et al. 2016b who used the macro-average of the F1-score for the stance labels' favor' and 'against' as the bottom-line evaluation metric. By taking the average of these two classes, they treat the 'neither' class as one which is not of interest. They argue that if one randomly accesses tweets, then the probability that one can infer 'favor' or 'against' stance towards a pre-chosen target of interest is small and hence has motivated the Information Retrieval-like metric used in the data set and across other SemEval tasks like sentiment prediction.The results for both classification task can be seen in 5.1 and 5.2

Model + Features	F1 macro - Average
Linear SVC + (1-3) n-grams	57.28
Linear SVC + Word2vec	57.04
Linear SVC + GloVe	55.98
Linear SVC + RoBERTa	57.13
Linear SVC + (1-3) n-grams + Hashtag	**60.61**
Linear SVC + Word2vec + Hashtag	58.52
Linear SVC + GloVe + Hashtag	57.71
Linear SVC + RoBERTa + Hashtag	59.77
Linear SVC + (1-3) n-grams + POS ratio + Hashtag	40.46
Linear SVC + Word2vec + POS ratio + Hashtag	40.47
Linear SVC + GloVe + POS ratio + Hashtag	36.83
Linear SVC + RoBERTa + POS ratio + Hashtag	42.40

Table 5.1: Table displaying the different results using different Hybrid Features settings. The best Result is that from using trigrams and Hashtag vectors.

Model	Abortion F1 macro avg	Atheism F1 macro avg	Climate F1 macro avg	Feminist Movement F1 macro avg	Hillary Clinton F1 macro avg	All F1 macro avg
Linear SVM ngrams (baseline)	**66.81**	62.75	48.17	58.21	58.88	59.91
Linear SVM + Word2vec	62.98	68.48	**57.25**	58.58	59.73	61.41
Linear SVM + Glove	61.28	64.06	47.96	61.32	58.84	59.92
Linear SVM + RoBERTa	63.46	**70.29**	47.96	**60.75**	**64.62**	**65.13**

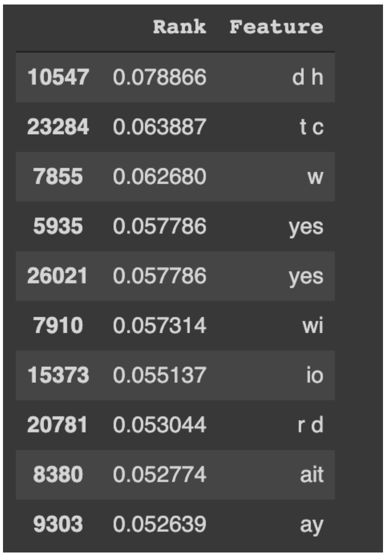
Table 5.2: Table showing the F1 scores of Linear SVC with different embeddings.The BERT model performed better across all target and had the best performance per target on Atheism.

Word Embeddings Performance

As shown from 5.1 and 5.2 the GloVe tends to perform slightly worse than Word2vec.This is unexpected as the pre-trained embedding used was trained on tweets, unlike word2vec, which is trained on a large corpus of text. The intuition here is that the dimension of the vector used may have accounted for this. I used 300 dimensions to represent each word with Word2vec against 200 dimensions in Glove, which is the maximum. More dimensions often mean a larger vector space to map information about the word. Although word2vec is trained on a completely different domain, Its vector space dimension could have given an edge in this situation.

Results from table ?? show that the baseline with the combination of (1-3) and (2-5) word and character n-grams seems to hold its weight performing only slightly worse than GloVe across all the targets and Hillary, performing best at Abortion. To further investigate this, I look into the n-grams, which account for the model's prediction output in this case. I use the coefficient values of the SVC model to classify the features according to their importance, as shown in figure 5.1.

Surprisingly, the words with the best features are mostly character n-grams. I expected unique words



	Rank	Feature
10547	0.078866	d h
23284	0.063887	t c
7855	0.062680	w
5935	0.057786	yes
26021	0.057786	yes
7910	0.057314	wi
15373	0.055137	io
20781	0.053044	r d
8380	0.052774	ait
9303	0.052639	ay

Figure 5.1: pandas frame showing the top 10 features by feature importance on the Linear SVM with (1-3) and (2-5) word and character ngrams for the target Hillary Clinton.

that would have helped in separating the values in the hyperplane. The reasoning for this could be that the sequence of characters is very repetitive, and the model learns to attribute higher weights to such values. Nevertheless, a look at the best results for the argument classification task from **??**in which I used a combination of (1-3) n-grams and bag of words of the set of hashtags, demonstrate that a combination of hashtags and full words were optimal as seen in 5.2

It can be seen that many hashtags contributed to the performance of the model such as *scotusmarriage menstruationmatters*,*abortiondemand*, *marriageequality*, *gaymarriage*, *baltimoreriots* and *centerhilter*. These Hashtags by themselves are very informative, in my opinion, as they sort of give a sense that the tweets attached to these are more likely to spark up debates resulting in arguments towards the target. An example tweet is stated below.

- *Target : Abortion*
  Tweet : There's an undeniable inverse correlation between women's right to choose and crime rates #womensrights #SemST

By visualising the frequency distribution of the hashtags using word cloud, we can see the appearance of some of those key hashtags that played an important role in the predictions as shown in 5.3.

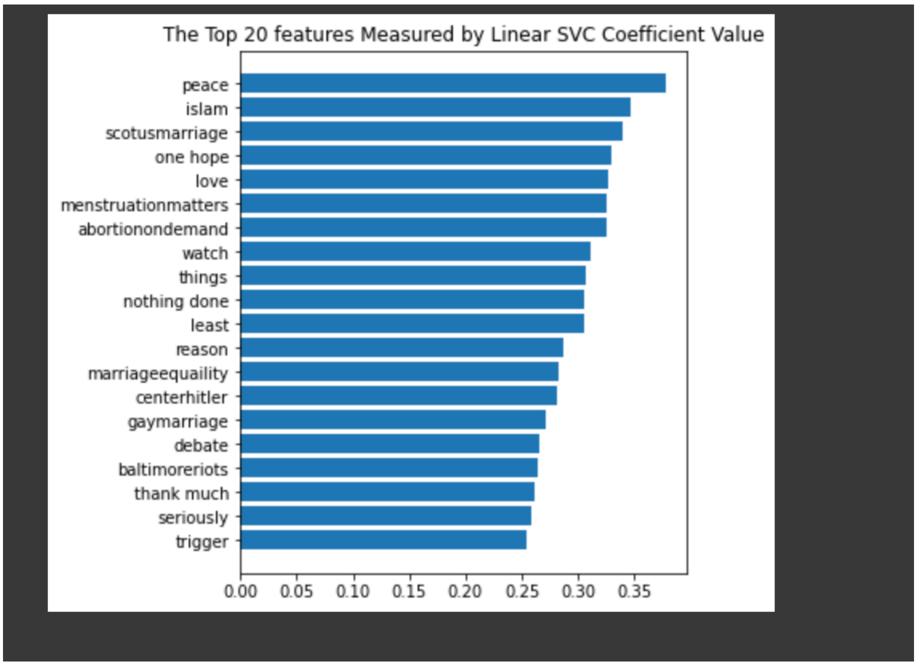


Figure 5.2: Bar chart showing the coefficient values of the Linear SVC for the features contributing more for distinguishing between argumentative and Non argumentative claims.The hashtags are the non separated words

Another point to notice is the unusual low scores for target like Climate change in 5.2.This can be as result of how particularly skewed the distribution of the against class are compared to the other classes as shown in 5.4.

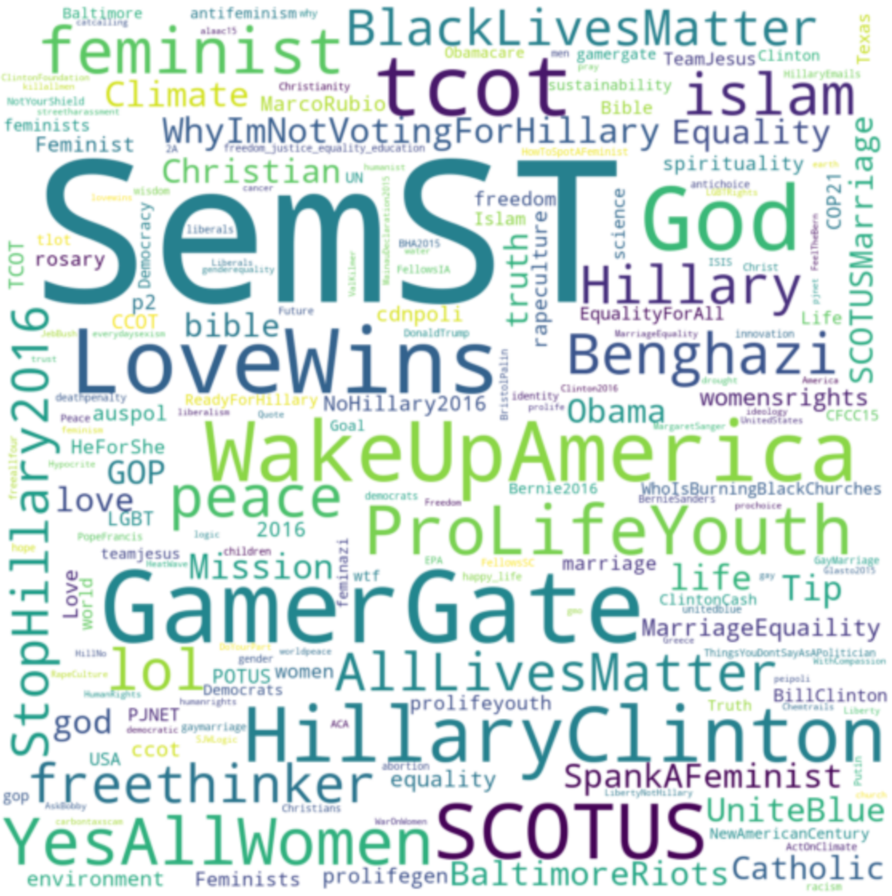


Figure 5.3: Word clouds showing distribution of hashtags in which we can discern debatable hashtags like *BlacklivesMatter*, *SCOTUSMarriage*, *MarriageEquality*, *womerights*

Noticeably ,It can be seen from 5.1 that including the POS ratio has worsen the results of the classifier.This can be as a results of the feature engineering step , where I stack the embeddings, hashtags and POS ratios on the horizontal plane.The embeddings and hashtags have a high dimension , so stacking it with discrete values of each POS ratio could cause a disproportionate weighing of the features when the model tries to fit the data set.

5.4.2 Extractive Summarization

I Evaluate the summaries generated by using the ROUGE metric as discussed above.Due to the high class imbalance in the target Climate change is a Real Concern, the summary evaluation is not assessed on it.Results are shown on 5.3

Summarizing Model	ROUGE-1 F1 Average	ROUGE-2 F1 Average	ROUGE-L F1 Average
SumBasic	39.91	12.76	29.21
TextRank	42.24	**17.00**	**32.80**
Word2vec + Clustering	38.52	11.67	28.56
GloVe + Clustering	40.23	14.57	30.88
RoBERTa + Kmedoid Clustering	**40.52**	13.15	29.71

Table 5.3: Table showing the Rouge scores of the different summarizing models. BERT embeddings followed by Kmedoids only achieve minute ROUGE-1 score over GloVe

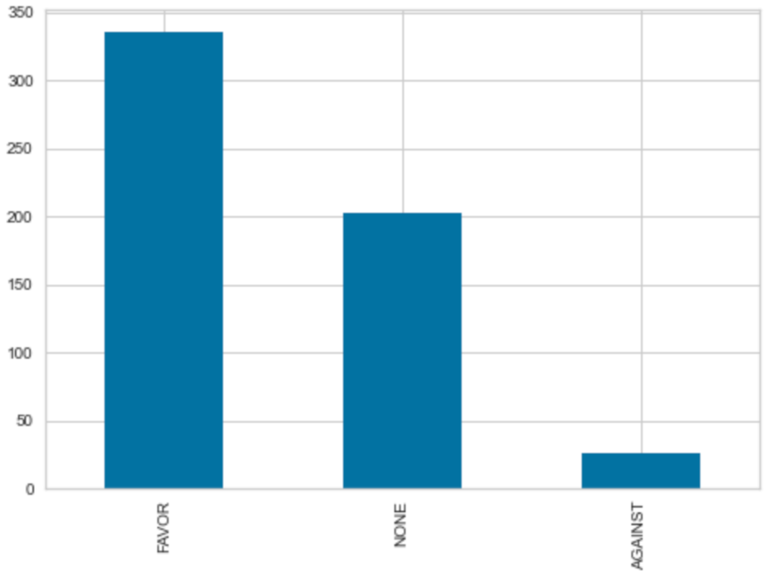


Figure 5.4: Distribution of labels on target Climate change is a Real Concern.

Unfortunately, The use of embeddings and clustering did not show any significant improvement over the baselines used. Implementing the clustering technique aimed at capturing the central tweets from the centroids with the intuition that they would capture the most salient aspect of a critical opinion, and the tweets clustered to them based on cosine similarity will provide similar critical information. However, algorithms like SumBasic, which relies primarily on the frequency distribution of the words across the corpus to select key sentences, and TextRank, which weighs the words based on graph-like structures, have the edge over the clustering method. Nevertheless, this does not necessarily mean that the approach is to be deemed unfit. One reason for this is that the summarization step relies on the 'ground truth' established by me. As a result, there is always the uncertainty that the references are not fully representative.

Unexpectedly, the BERT embeddings provide only slightly better results than the Word2vec and GloVe embeddings. However, in a task as summarisation, we would expect it to perform higher as compared to these word embeddings as it is able to learn representation at the sentence level. Further, we would expect BERT to at the very least score the best on ROUGE-L for the longest sequences of text, but it is not the case. It should be known that the ROUGE metric is not great at evaluating sentences at the semantic level. It can therefore be argued that the BERT embeddings are actually undervalued by the metric, especially when it comes to the ROUGE-L scores.

Below is a view of the first 4 tweets selected from the summaries generated by the models and the gold standard summaries for some of targets and stance labels.

Target:Legalization of Abortion, Stance: Against

Gold Standard summary

- True progressives work toward greater inclusion and protections for the marginalized. Standing up for the unborn is progressive!
- I really don't understand how some people are Pro-Choice. A life is a life no matter if it's 2 weeks old or 20 years old. #SemST
- RT @GrumpyOldGuy2: #DebbieWashermanSchultz the woman has a voice the doctor has a voice. Who speaks for the baby? I'm just askin. #SemST

- Children are the greatest blessing which God has bestowed on man and woman. -Pope Francis #LifeisaGift #SemST

SumBasic

- @crazygranny56 More #blacks killed by abortion than during #slavery #ConfederateFlag #SemST
- So can unborn children have rights now? #SemST
- RT @RogerIsCatholic: Save the Baby Humans! #WAAR #SemST
- Before we go looking for #life on other planets we should #stop #killing life on this one. #prolifeyouth #prolifegen #SemST
- I can tell women what to do with their bodies #ThingsYouDontSayAsAPolitician #SemST

RoBERTa + K-medoid Clustering

- @SenSchumer You sponsored the religious freedom bill. What is this bill that allows killing full term babies in NY? #tcot #SemST
- If #BlackLivesMatter why not black babies in the womb? Somehow their lives don't matter.... puzzled with that one. #catholic #SemST
- @GBPstaff 1999 Meet the Press admitted to being Very Pro-Choice even with late term Partial Birth Abortions. He's sick! #SemST
- Why don't black lives matter in the womb? #SemST

Word2vec + K-medoid Clustering

- Also, abortion is wrong biblically and morally. Imagine never getting a chance to live your life. Think about where you are now. #SemST
- @hillaryclinton - I guess this means you're happy for me to be pro-life as I never aborted any of my own children? #hypocrisy #SemST
- Just like abortion, even though SCOTUS says gay marriage is legal doesn't make it moral. #SCOTUSMarriage #FreedomOfReligion #SemST
- Idiot: how would you feel if your mom aborted you? Me: nothing cause I would be fucking dead. #SemST

GloVe + K-medoid Clustering

- @notmuchelse But of course today one doesn't have to be responsible You can kill your child instead! #SemST
- We are people who believe every child is entitled to life and liberty.... -@BarackObama Yet abortion is still legal. #SemST
- if we can accept that a mother can kill even her own child, how can we tell other people not to kill one another? #SemST
- You support abortion? Never had an abortion? You'll still have to give account to your Creator for the taking of innocent life. #SemST

TextRank

- Planned Parenthood lines its pockets while sacrificing women and children in need at the altar of abortion.
- I will agree to gay marriage if you agree to pro-life and then adopt babies that would have been aborted.
- Why is it illegal to kill an unhatched eagle egg but its legal to kill an unborn human baby?isright
- Hey Megan–a baby's heart starts beating at 21 days after conception. No time is the right time to end its life!! #kellyfile #SemST

Target: Feminist movement, Stance: Favor

Gold Standard summary

- Job should always go to best candidate, regardless of gender. Gender shouldn't even matter anymore, it's 2015! #PaulHenry #SemST
- Did you know? Gender stereotypes as we know then developed with beginning of the 18. century. #gender #SemST
- You know you're in a patriarchy when women are the ones who are blamed for prostitution #whatisjustice #inequality #SemST
- A woman is not a sex object. She's a person. #truth #wisdom #womensrights media life humanity #love #society #SemST

SumBasic

- @PH4NT4M @MarcusChoOo @CheyenneWYN women. The term is women. Misogynist! #SemST
- People think because I'm a feminist I hate men. No I'm a feminist, I believe in equality for all. #EqualityForAll #SemST
- @njaniboy99 @omercayir1 @Sahil_Handa_ i don't ever need a man #YesAllWomen #SemST
- @Mike_Charmander @MsLatina a right doesn't have to come from a law. They're called equal rights not equal laws #SemST

TextRank

- tired of anti-womanist respectability politics. we need a safe space for women can look like they're at the club at an interview #SemST
- It upsets me how people are so narrow-minded when it comes to feminism. STEREOTYPING MEN IS NOT . #SemST
- when they say men look at women like a piece of meat what do they even mean, they want to cook eat her? #YesAllWomen #SemST
- Thanks to the work of people like @everydaysexism they are not only helping women but helping the whole of society #SemST

Word2vec + K-medoid Clustering

- You want to hear something really ugly, 1 in 5 women will be sexually assaulted in their time in college #MisogynyIsUgly #SemST

- i wish people would understand that true feminists want equality for everyone (even men) not just girls . #SemST
- @randomtweetor i hate to break it to u bruh but women do get pretty for us. They get pretty to show other women #feminist #SemST
- I don't understand how the concept of a male being a feminist doesn't get in some peoples heads. Equality? Anyone? #feminist #SemST

GloVe + K-medoid Clustering

- I guess wanting and trying to make this world a better place for everyone makes me ugly. #feminist #SemST
- RT @LZats: Do you know what women should stop wearing after age 30? Nothing. Women can wear whatever they want, no matter how old. #SemST
- i wish people would understand that true feminists want equality for everyone (even men) not just girls . #SemST
- I dont need white privillaged men coming up to me telling me my beliefs and fashion are unacceptable furry #Feminist #SemST

RoBERTa + K-medoid Clustering

- tired of anti-womanist respectability politics. we need a safe space for women can look like they're at the club at an interview #SemST
- Yes women it's not fair that men staying at home is shamed by society. We must change ppl additudes #everydaysexism SemST
- the fact that people think women need to be appealing and attractive to be heard/seen as equal is why we need feminism #SemST
- @moo_mena unlearn that believing in gender equality is life saving, yes it's fab-but why not actively fight for change? #SemST

Although each component is addressed individually, the general comments can be made based on the final summaries produced when combining all components. Because of the size of the data set and class imbalance issues, for some targets, tweets extracted after the argument classification and stance detection step were too small in size to be clustered effectively(only generates 1 cluster). As such, the extracted tweets after stance detection were treated as the summary. An example is that of tweets that are in favor of the target *Hillary Clinton* with only 11 tweets as shown below.

- Congratulations to our Women Soccer Team for just winning the World Cup against Japan 5-2, women rule ! #freeallfour #SemST
- @HillaryClinton: Here's to fearless women chasing their goals. Congratulations, Team #USA! H #SemST
- @AdamSmith_USA because clearly @HillaryClinton is a champion for us all. #SemST
- @HillaryClinton the @DalaiLama speaks of women in leadership roles bringing about a more compassionate world. #potus #SemST
- Have full faith in @HillaryClinton's campaign. #strategy #politics #SemST
- @HillaryClinton is plain amazing, humble, smart and confident! Let's get it Mrs. President! #HillaryClinton #SemST
- President Jimmy Carter: There's no doubt Hillary will get the nomination. And when she does I'll be happy to support her #SemST
- @thehill : Women deserve a better candidate for the HIGH HONOR if first woman President: We ALL do! #WhyI'mNotVotingForHillary #SemST
- @HillaryClinton : Women deserve a better candidate for the HIGH HONOR of first woman President. We ALL do! #SemST
- Enjoyed @jamiaw article on feminism + @hillaryclinton.We are building campaign that engages ppl through an intersectional lens #SemST
- Proud of @HillaryClinton for supporting stronger gun control measures. #ReadyForHillary #SemST

5.5 Comparison with Other Work

In this section I compare some of the results against proposed works in similar or different domains.

Comparison with SemEval author's model on tweets expressing opinion towards the target

The best results obtained for the stance detection outperforms the model produced by the task organizers on tweets that expressed an opinion towards a target. I further compare my model's performance solely on the stance detection task (no prior knowledge of opinions expressed towards the target) to that of other teams submissions in the SemEval shared task Mohammad et al. 2016b.The result can be seen on 5.4.

Model	F1- Macro Average
Linear SVC + RoBERTa	**65.13**
SVM Classifier + n-gams + Target Mohammad et al. 2017	62.5

Table 5.4: Table showing my model performance and that of Mohammad et al. 2017 on Tweets that expressed an Opinion towards a target.Comparison of F1 average scores on all targets.

My model was able to achieve higher performance without using the presence or absence of the Target in the tweets as a feature like that presented by the task organizers. The BERT embeddings have been able to capture a more meaningful semantic relationship between the words in the sentences and mapped it to the target variables.

Team Results Comparison

Table 5.5 shown below displays the performance of my model on the stance detection task as stated in comparison to other teams in the competition.

Participating Teams	F1 - Macro average
MITRE Zarrella and Marsh 2016	56.02
pkudblab Moens et al. 2007	**58.59**
Takelab Tutek et al. 2016	58
ECNU Zhang and Lan 2016	55.72
IUCL-RF Liu et al. 2016	51.5
CU-GWU Elfardy and Diab 2016	51.90
DeepStance Vijayaraghavan et al. 2016	52.86
My model (Linear SVC + RoBERTa)	**55.26**

Table 5.5: Results of Some of the Team's model performance on Task A of the SemEval 2016 Stance Detection Task Mohammad et al. 2016b.

Argument Classification Comparison

The Linear SVC with n-grams and hashtags features for argument classification developed has outperformed the BERT model pre-trained on the UKP Corpus Stab et al. 2018 by Reimers et al. 2019 as demonstrated on 5.6.

Model	F1- Macro Average
Linear SVC + n-grams + Hashtags	**60.61**
BERT-base(topics) Reimers et al. 2019	49.81

Table 5.6: Table showing F1- Macro average on my argument classifier and the fine-tuned BERT Classifier.

The results shown emphasizes how state of the art models for argument classification may not only be too complex to develop and computationally expensive but also not fit to achieve best results on the type of text presented in Twitter, motivating the use for simpler modeling structures as can be seen with a combination of word n-grams, and metadata features like Hashtags.

5.6 Limitations

5.6.1 Interpretability

Words and sentence embeddings are challenging to disseminate into individual features mainly due to the fact that words are represented in a highly dimensional vector space, making it complex to trace back to the original word. However, based on the proceedings of BERT as compared to the word embeddings like Word2vec and Glove, it can be argued as to why it manages to obtain the best results across all targets.

- Word2vec and GloVe models generate embeddings that are context-independent. There is only one vector representation for each word.Propose an example.

- Word2vec and GloVe can not generate representations of words outside of their vocabulary.On the other hand, BERT learns representations at sub words which are words represented between character and word embeddings, reducing its vocabulary but making it able to process out of vocabulary words.

5.6.2 References

One of the main limitations of this project is the ground truths generated to evaluate the summaries. Professional annotators should ideally construct references as they would have the expertise and domain knowledge required. Self-creating the summaries brings about a level of uncertainty regarding its total validation and confidence that I am testing against the right target. In addition to this, ROUGE also comes with some drawbacks. It does not cater to different words that have the same meaning as it measures syntactical matches rather than semantics. Therefore, if I have two sequences that have a similar context but use different words to express that, then the ROUGE scores will below. This can be counter-intuitive for sentence embeddings obtained with BERT, which get the vector representation of the sentence as a whole and aims at capturing the semantic discrepancies.

An alternative to the ROUGE metric is the sentence level fluency evaluation work by Kann et al. 2018. It proposes a syntactic log-odds ratio (SLOR) which is a a normalized language model score, as a metric for reference-less fluency evaluation of natural language generation output at the sentence level removing the need for human annotated summaries.

5.6.3 Target-specific Classification

Another major issue with the approach taken in this paper is that the classifications are performed on known pre-existing targets and hence can not perform in an open domain target setting. I implemented the procedure in creating training examples of claims and annotations for target by Bar-Haim et al. 2017 in their open domain target identification process. Unfortunately, tweets, unlike articles and news, may not always contain a target, subject, or topic of discussion. This was the case as the data set created from this was severely imbalanced with only three training examples that contained a target.

Chapter 6

Conclusion

In this chapter, I account for the main contributions, achievements and describe possible future work outcomes.

6.1 Reflections

In this paper, I explore the use of a three-component system made of arguments classification, stance detection, and Extractive summarization to extract the most meaningful claims that demonstrate the opinions of a user supporting or opposing a target. Different experimentation with embeddings and metadata features was done to build the classifiers. Furthermore, I developed an algorithm to select the key tweets based on k-medoids clustering with cosine distance.

Concerning the aims and objectives described in the paper, I was able to implement an argument classifier and a stance detection model, which showed promising results over the baselines that were set and through a comparison with other work such as the fine-tuned BERT model from Reimers et al. 2019 and the participating teams in the SemEval shared task. In addition, I have annotated the data set with human-created gold standard summaries to allow the evaluation of the extractive summarization step.

The clustering of tweets has brought disappointing results when it comes to the final extracted summaries. Through the summarization step, I discovered that clustering the word vectors to extract meaningful claims does not provide satisfying results over the baselines algorithms that make use of frequency distributions and tree structures to extract the key sentences. In addition, a limit in the size of the data set poses a constraint as to whether clusters would be generated, as discussed in the evaluation. Approaching the Extractive Summarization step through clustering has shown not to be ideal.

6.2 Discussion and Future Work

This project has been focused on developing models for target-specific entities. As a result, because the target is known by the model, it can not be used in an open setting. An interesting area of research will be open target identification to identifying the target in the tweets. I also posed the assumption that conversations and debates that are controversial on Twitter will contain arguments. Something I did not consider is what aspect of a tweet makes it controversial. My intuition here is that a successful attempt at classifying those tweets that are controversial will help in separating claims that have a structured comment about a target, improving results in target identification.

In this paper , I focused on modelling with different embeddings and meta data.A further step to take perhaps would be the use of Recurrent Neural Networks(RNN) and its variant LSTM(Long Short-term memory).Essentially text data is read continuously from one direction to another making it sequential in nature.This attribute could be of benefit to model than progressively read the text in this way.Neural networks like RNNs are specialized in working with such data due to their architecture which allows them to have a temporal sequence in which they can remember the inputs that was feed into it.

A more specialised variant would be the LSTM.RNNs can only remember so much about the input that was feed into it and so it becomes problematic when dealing with longer sequences of text. LSTMs approach provides a remedy to this by continuously letting go of irrelevant context and only remembering the context which is needed to solve the problem given.Convolutional Neural Networks(CNN) has also proven to be a great approach when it comes to text classification.Each word in a sentence is replaced by its word vector representation and they all of the same dimension.Using the direct approach of convolution when it comes to images, the word vectors can be stacked up together to form a matrix of size n X d where n is the number of words in the sentence and d is the size of the word vectors.CNNs have the ability to look at group of words together using a context window similarly to how Word2vec embeddings are trained.This gives it the ability to look at different neighbouring words when making sense of a sentence during classification.

Regarding the meta data features that were used, It would be interesting to experiment with more of them.The Data set restricted me to only using the tweets , but Twitter API's can provide much more features such as the user's retweets,likes, their followers and people that follow them.We saw that hashtags contributed a lot to the performance of the models especially in cases where tweets were mainly written with those. It would be compelling to investigate how these other meta data features can play in the classification decisions.

There is a limitation when it comes to languages in which the model excels.It would undoubtedly perform poorly if executed on different languages like French and Spanish for example.There has been a severe lack of work done when it comes to the research on minority languages as well. Furthermore, I previously talked about the issue of multilingual tweets, where more than one language is expressed in the text.This is often a recurrent issue seen in social media.I believe research on social media data translations and interpretations is one in which attention should be pointed.

The nature of the tweets has made it a very challenging task when it came to the classifications and summarization components.The presence of the short text poses a problem to how much contextual information the model can actually infer.Teams that submitted results on the stance detection using architectures like RNNs and LSTMs which benefit from sequential data did not beat the baseline that was set by the task organizers with a Linear SVC.This can perhaps motivate the application of this model on other social media platforms which are not necessarily restricted by text length such as Reddit and Facebook and can see and major improvement.

Bibliography

Aseel Addawood and Masooda Bashir. “what is your evidence?” a study of controversial topics on social media. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 1–11, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-2801. URL https://www.aclweb.org/anthology/W16-2801.

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.

Leila Amgoud, Florence Bannay, Charlotte Costedoat, Patrick Saint-Dizier, and Camille Albert. Introducing argumention in opinion analysis: Language and reasoning challenges. In *Proceedings of the Workshop on Sentiment Analysis where AI meets Psychology (SAAIP 2011)*, pages 28–34, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing. URL https://www.aclweb.org/anthology/W11-3705.

Roy Bar-Haim, Indrajit Bhattacharya, Francesco Dinuzzo, Amrita Saha, and Noam Slonim. Stance classification of context-dependent claims. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 251–261, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E17-1024.

Farah Benamara, Maite Taboada, and Yannick Mathieu. Evaluative language beyond bags of words: Linguistic insights and computational applications. *Computational Linguistics*, 43(1):201–264, April 2017. doi: 10.1162/COLI_a_00278. URL https://www.aclweb.org/anthology/J17-1006.

Benjamin Bengfort, Rebecca Bilbro, Nathan Danielsen, Larry Gray, Kristen McIntyre, Prema Roman, Zijie Poh, et al. Yellowbrick, 2018. URL http://www.scikit-yb.org/en/latest/.

James C. Bezdek, Robert Ehrlich, and William Full. Fcm: The fuzzy c-means clustering algorithm. *Computers Geosciences*, 10(2):191–203, 1984. ISSN 0098-3004. doi: https://doi.org/10.1016/0098-3004(84)90020-7. URL https://www.sciencedirect.com/science/article/pii/0098300484900207.

DOUGLAS BIBER and EDWARD FINEGAN. Styles of stance in english: Lexical and grammatical marking of evidentiality and affect. *Text - Interdisciplinary Journal for the Study of Discourse*, 9(1):93–124, 1989. doi: doi:10.1515/text.1.1989.9.1.93. URL https://doi.org/10.1515/text.1.1989.9.1.93.

Tom Bosc, Elena Cabrio, and Serena Villata. DART: a dataset of arguments and their relations on Twitter. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 1258–1263, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL https://www.aclweb.org/anthology/L16-1200.

bsolomon1124. bsolomon1124/demoji. URL https://github.com/bsolomon1124/demoji.

Roshni Chakraborty, Maitry Bhavsar, Sourav Kumar Dandapat, and Joydeep Chandra. Tweet summarization of news articles: An objective ordering-based perspective. *IEEE Transactions on Computational Social Systems*, 6(4):761–777, 2019. doi: 10.1109/TCSS.2019.2926144.

Chetan Chavan and Ranjeetsingh Suryawanshi. Summarization of tweets and named entity recognition from tweet segmentation. In *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, pages 66–71, 2016. doi: 10.1109/ICACDOT.2016.7877553.

Kareem Darwish, Peter Stefanov, Michaël J. Aupetit, and Preslav Nakov. Unsupervised user stance detection on twitter. *CoRR*, abs/1904.02000, 2019. URL http://arxiv.org/abs/1904.02000.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

Heba Elfardy and Mona Diab. CU-GWU perspective at SemEval-2016 task 6: Ideological stance detection in informal text. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 434–439, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/S16-1070. URL https://www.aclweb.org/anthology/S16-1070.

Lorenzo Ferrone and Fabio Massimo Zanzotto. Symbolic, distributed, and distributional representations for natural language processing in the era of deep learning: A survey. *Frontiers in Robotics and AI*, 6, Jan 2020. ISSN 2296-9144. doi: 10.3389/frobt.2019.00153. URL http://dx.doi.org/10.3389/frobt.2019.00153.

J. Firth. A synopsis of linguistic theory 1930-1955. In *Studies in Linguistic Analysis*. Philological Society, Oxford, 1957. reprinted in Palmer, F. (ed. 1968) Selected Papers of J. R. Firth, Longman, Harlow.

Shai Gretz, Roni Friedman, Edo Cohen-Karlik, Assaf Toledo, Dan Lahav, Ranit Aharonov, and Noam Slonim. A large-scale dataset for argument quality ranking: Construction and analysis, 2019.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020. URL https://doi.org/10.5281/zenodo.1212303.

J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

Myungha Jang and James Allan. Explaining controversy on social media via stance summarization. In *The 41st International ACM SIGIR Conference on Research Development in Information Retrieval*, SIGIR '18, page 1221–1224, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356572. doi: 10.1145/3209978.3210143. URL https://doi.org/10.1145/3209978.3210143.

Katharina Kann, Sascha Rothe, and Katja Filippova. Sentence-level fluency evaluation: References help, but can be spared! In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 313–323, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/K18-1031. URL https://www.aclweb.org/anthology/K18-1031.

N. Kaoungku, K. Suksut, R. Chanklan, K. Kerdprasop, and Nittaya Kerdprasop. The silhouette width criterion for clustering and association mining to select image features. *International Journal of Machine Learning and Computing*, 8:69–73, 02 2018. doi: 10.18178/ijmlc.2018.8.1.665.

Keredson. keredson/wordninja. URL https://github.com/keredson/wordninja.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-1013.

Can Liu, Wen Li, Bradford Demarest, Yue Chen, Sara Couture, Daniel Dakota, Nikita Haduong, Noah Kaufman, Andrew Lamont, Manan Pancholi, Kenneth Steimel, and Sandra Kübler. IUCL at SemEval-2016 task 6: An ensemble model for stance detection in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 394–400, San Diego, California, June 2016.

Association for Computational Linguistics. doi: 10.18653/v1/S16-1064. URL https://www.aclweb.org/anthology/S16-1064.

Yang Liu and Mirella Lapata. Text summarization with pretrained encoders, 2019.

Yang Liu, Ivan Titov, and Mirella Lapata. Single document summarization as tree induction. 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 1745–1755, Minneapolis, Minnesota, June 2019a. Association for Computational Linguistics. doi: 10.18653/v1/N19-1173. URL https://www.aclweb.org/anthology/N19-1173.

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, 2019b.

Edward Loper and Steven Bird. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*, 2002.

Wes McKinney et al. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX, 2010.

Rada Mihalcea and Paul Tarau. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-3252.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality, 2013.

Miso-Belica. miso-belica/sumy. URL https://github.com/miso-belica/sumy.

Marie-Francine Moens, Erik Boiy, Raquel Mochales, and Chris Reed. Automatic detection of arguments in legal texts. pages 225–230, 01 2007. doi: 10.1145/1276318.1276362.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. A dataset for detecting stance in tweets. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3945–3952, Portorož, Slovenia, May 2016a. European Language Resources Association (ELRA). URL https://www.aclweb.org/anthology/L16-1623.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. SemEval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41, San Diego, California, June 2016b. Association for Computational Linguistics. doi: 10.18653/v1/S16-1003. URL https://www.aclweb.org/anthology/S16-1003.

Saif M. Mohammad, Parinaz Sobhani, and Svetlana Kiritchenko. Stance and sentiment in tweets. 17(3), June 2017. ISSN 1533-5399. doi: 10.1145/3003433. URL https://doi.org/10.1145/3003433.

Siddhi Naik, Shruti Lade, Swati Mamidipelli, and Ashwini Save. Tweet summarization: A new approch. In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pages 1022–1025, 2018. doi: 10.1109/ICICCT.2018.8473327.

A. Nenkova and Lucy Vanderwende. The impact of frequency on summarization. 2005.

Layla Oesper, Daniele Merico, Ruth Isserlin, and Gary D Bader. Wordcloud: a cytoscape plugin to create a visual semantic summary of networks. *Source code for biology and medicine*, 6(1):7, 2011.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

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.

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

Pltrdy. pltrdy/rouge. URL https://github.com/pltrdy/rouge.

Radim Rehurek and Petr Sojka. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2), 2011.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084, 2019. URL http://arxiv.org/abs/1908.10084.

Nils Reimers, Benjamin Schiller, Tilman Beck, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. Classification and clustering of arguments with contextualized word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 567–578, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1054. URL https://www.aclweb.org/anthology/P19-1054.

Swapna Somasundaran and Janyce Wiebe. Recognizing stances in ideological on-line debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 116–124, Los Angeles, CA, June 2010. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W10-0214.

Christian Stab, Tristan Miller, Benjamin Schiller, Pranav Rai, and Iryna Gurevych. Cross-topic argument mining from heterogeneous sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3664–3674, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1402. URL https://www.aclweb.org/anthology/D18-1402.

Matt Thomas, Bo Pang, and Lillian Lee. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 327–335, Sydney, Australia, July 2006. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W06-1639.

Stephen E. Toulmin. *The Uses of Argument*. Cambridge University Press, July 2003. ISBN 0521534836. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike-20&path=ASIN/0521534836.

Martin Tutek, Ivan Sekulić, Paula Gombar, Ivan Paljak, Filip Čulinović, Filip Boltužić, Mladen Karan, Domagoj Alagić, and Jan Šnajder. TakeLab at SemEval-2016 task 6: Stance classification in tweets using a genetic algorithm based ensemble. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 464–468, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/S16-1075. URL https://www.aclweb.org/anthology/S16-1075.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.

Prashanth Vijayaraghavan, Ivan Sysoev, Soroush Vosoughi, and Deb Roy. Deepstance at semeval-2016 task 6: Detecting stance in tweets using character and word-level cnns, 2016.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Wei Xu, Ralph Grishman, Adam Meyers, and Alan Ritter. A preliminary study of tweet summarization using information extraction. In *Proceedings of the Workshop on Language Analysis in Social Media*, pages 20–29, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W13-1103.

Guido Zarrella and Amy Marsh. MITRE at SemEval-2016 task 6: Transfer learning for stance detection. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 458–463, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/S16-1074. URL https://www.aclweb.org/anthology/S16-1074.

Zhihua Zhang and Man Lan. ECNU at SemEval 2016 task 6: Relevant or not? supportive or not? a two-step learning system for automatic detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 451–457, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/S16-1073. URL https://www.aclweb.org/anthology/S16-1073.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–663, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1061. URL https://www.aclweb.org/anthology/P18-1061.