ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ

ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS

Department of Management Science and Technology

Master of Science in Business Analytics FT 2020-2021

Argumentation Mining – Pilot Project

Course: Machine Learning & Content Analytics

Professors: Haris Papageorgiou

George Perakis

Aris Fergadis

Authors: Lamaj Kostandin (f2822017)

Vaiou Konstantinos (f2822002)

Table of contents

1.1 Introduction 3

1.2 Our Project 3

1.3 Our Goals 3

2.1 Methodology 3

2.2 Data Collection 4

2.3 Data Overview 4

2.4 Data Processing / Annotation 6

2.5.1 Baseline Algorithm for argument prediction 8

2.5.2 Baseline Algorithm for structure prediction 10

2.5.3 Fasttext architecture 10

3.1.1 Fasttext configuration for Argument Mining 12

3.1.2 Fasttext configuration for Structure Mining 14

3.2. Results & Quantitative analysis 15

3.3 Qualitative & Error analysis 24

3.4 Future Work 25

4.1 Members 25

4.2 Time Plan 25

4.3 Bibliography 26

4.4 Appendices 26

1.1 Introduction

The scope of this project is to create robust models for argument & structure prediction of sentences in abstracts of scientific papers. Moreover, the knowledge obtained from the models is used for clustering similar abstracts.

1.2 Our project

In more detail, our team labeled sentences of 100 scientific abstracts with regards to their arguments (Neither, Evidence, Claim) and to their Structure (None, Background, Objective, Method, Results, Conclusion). This labeled dataset was combined with 900 other abstracts that were labeled from our fellow colleagues to create a completed dataset of 1017 abstracts. Moreover, for the purpose of argument mining an additional dataset with 1669 abstracts was provided.

These data were fed to a fasttext model for prediction and compared to our baseline models. Then the embeddings previously calculated were used to measure similarity between abstracts and their relationships were visualized in a graph.

1.3 Our goals

The purpose of these endeavor was to create a baseline approach for argumentation mining and word embeddings to be used in other models.

2.1 Methodology

To be able to evaluate our more complex models we first created two baseline models that we would compare them to. The first one is related to argument prediction and the second one to structure prediction.

For the baseline model for argument prediction, we split the data to train, validation and test. The approach used was that of a lexicon. In more detail, three different vocabularies were created from the training dataset by extracting tokens with specific techniques. The first one contained all tokens of the train dataset, the second one only the tokens from sentences labeled as evidence and the third one only the tokens from sentences labeled as claim. The set difference of the 50 most encountered tokens of these vocabularies was used for validating 3 different models respectively. The three different approaches corresponded to the number of tokens that was going to used from these sets for prediction. After evaluating and concluding on the best one the top 5 tokens in the sentences labeled as evidence and the corresponding ones for the sentences labeled as claim were used to test our baseline model.

On the other hand, for our baseline model for structure prediction we used the distribution of the labels in the abstract as our predictor. In this case we split the data to train and test to random abstracts without replacement. Then the train dataset was used to create a dictionary of the labels encountered in sentences of an abstract, for abstracts of the same size (same number of sentences). The most encountered label was used for prediction particular sentences of abstracts of the same size.

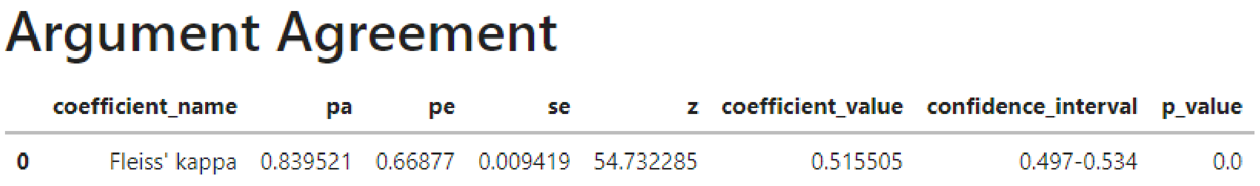
For our more complex approach for argument and structure mining we used the fasttext approach. Particularly, the dataset was converted to a format that the fasttext model is able to read and was split to train, validation and test with stratified sampling. This was used as input to the fasttext model and autotune to the validation set, so the best hyperparameters could be estimated.

Finally, the argument embeddings previously calculated were used for measuring abstract similarity between abstracts that contained sentences labeled as evidence or claim. The cosine similarity was used to create a graph of these documents and visualize their communities. It is important to note that edges on the graph were created only to documents that were included in the top 5 closest neighbors of an abstract.

Evaluation metrics that we used for comparing these models include but are not limited to accuracy, precision, recall, f1-score, confusion matrices and average loss plots.

2.2 Data collection

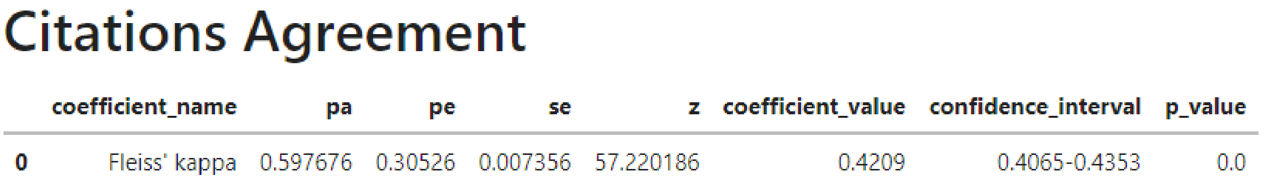
To begin with, the datasets include scientific abstracts linked with the Sustainable Development Goals of the United Nations. More specifically, our data include the dataset.json, dataset_aueb_argument_v3.json, dataset_aueb_structure_v2.json, dataset_aueb_citations_v2.json and eu_calls.json files. What we need to clarify explicitly is that the dataset.json archive was given to us from the Project Instructor, so as to feed our machine learning models. The rest archives include information related to the arguments, citations, structure and objective of the abstracts that all the teams had to read and annotate accordingly. In our case, our team had to annotate 100 abstracts to their corresponding labels with respects to arguments, structure and citations. Moreover, we must bear in mind that a higher agreement score between annotators corresponds to more realistic results for our models and a more meaningful application of our findings. Below we can see the scores for each datasets agreement.



Argument Agreement

	coefficient_name	pa	pe	se	z	coefficient_value	confidence_interval	p_value
0	Fleiss' kappa	0.839521	0.66877	0.009419	54.732285	0.515505	0.497-0.534	0.0

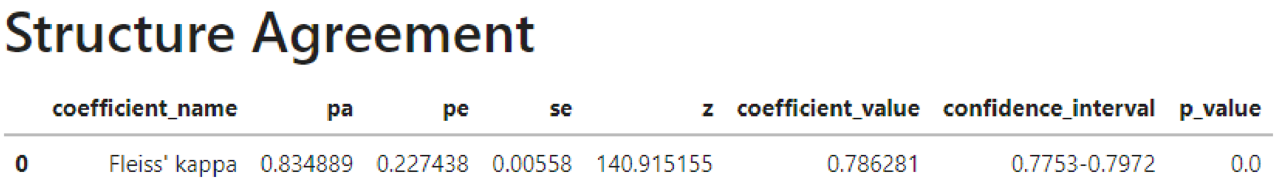
Figure 1: Argument Agreement



Citations Agreement

	coefficient_name	pa	pe	se	z	coefficient_value	confidence_interval	p_value
0	Fleiss' kappa	0.597676	0.30526	0.007356	57.220186	0.4209	0.4065-0.4353	0.0

Figure 2: Citations Agreement



Structure Agreement

	coefficient_name	pa	pe	se	z	coefficient_value	confidence_interval	p_value
0	Fleiss' kappa	0.834889	0.227438	0.00558	140.915155	0.786281	0.7753-0.7972	0.0

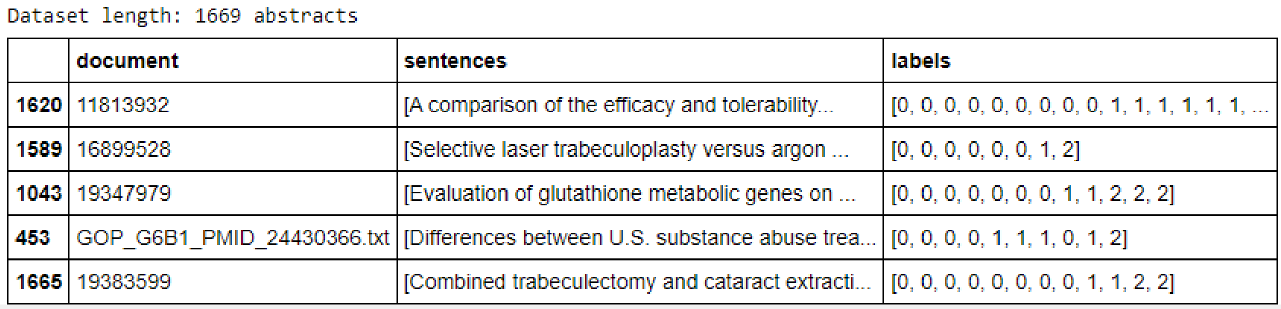
Figure 3: Structure Agreement

2.3 Data Overview

The dataset.json file contains information on the DOI[1] code of an abstract , the SDG[2] number, the title of the abstract, its contents and labels with regards to argument prediction (None, Evidence, Claim). Its contents except for the SDG number were used for argument mining purposes. In addition, it includes 1669 abstracts or 21429 sentences. Of these sentences 14548 are labeled

[1] DOI: Digital Object Identifier
[2] SDG: Sustainable Development Goals

None, 4510 are labeled evidence, 2371 are labeled claim. Before loading the data labels were mapped as None:0, Evidence:1 ,Claim:2 for consistency between datasets.

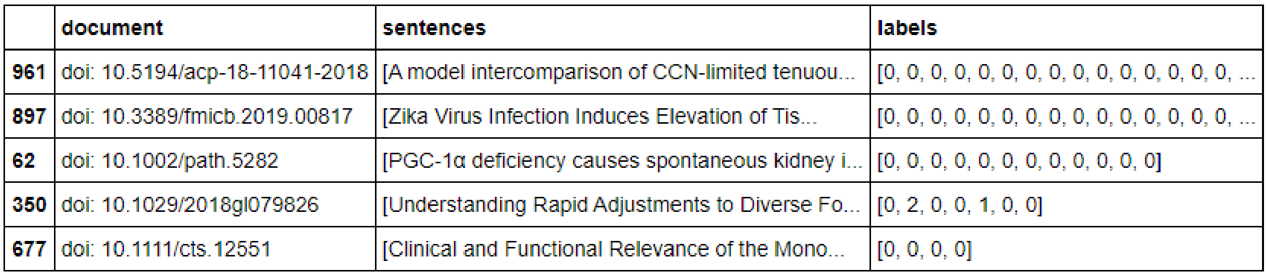


```
Dataset length: 1669 abstracts
```

	document	sentences	labels
1620	11813932	[A comparison of the efficacy and tolerability...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, ...
1589	16899528	[Selective laser trabeculoplasty versus argon ...	[0, 0, 0, 0, 0, 0, 1, 2]
1043	19347979	[Evaluation of glutathione metabolic genes on ...	[0, 0, 0, 0, 0, 0, 0, 1, 1, 2, 2, 2]
453	GOP_G6B1_PMID_24430366.txt	[Differences between U.S. substance abuse trea...	[0, 0, 0, 0, 1, 1, 1, 0, 1, 2]
1665	19383599	[Combined trabeculectomy and cataract extracti...	[0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 2, 2]

Figure 4:dataset for argument I

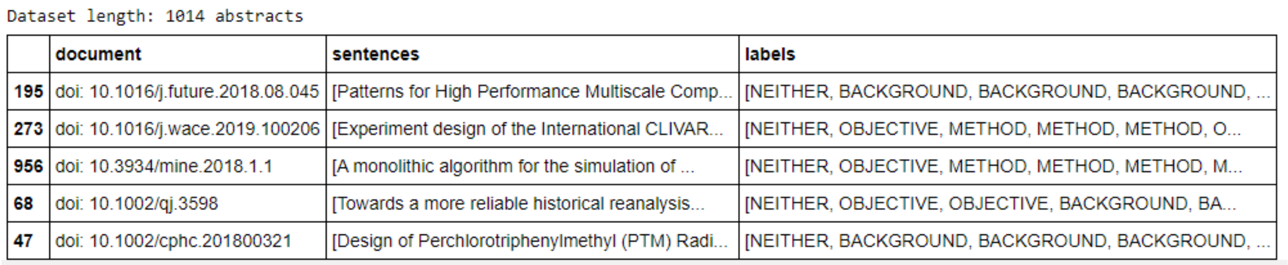
The dataset_aueb_argument_v3.json file among the things previously mentioned it includes the EU call identification, the project's objective, the URL to the abstract and the team that performed the annotation. As in the previous case the first sentence of each abstract corresponds to its title. Similarly, only information with regards to DOI, sentences and labels was kept for our study. The dataset contains 1017 abstracts or 10575 sentences. 7827 of the sentences are labeled Neither, 1700 are labeled evidence and 1048 are labeled claim. Before loading the data labels were mapped as Neither:0, Evidence:1 ,Claim:2 for consistency between datasets.



	document	sentences	labels
961	doi: 10.5194/acp-18-11041-2018	[A model intercomparison of CCN-limited tenuou...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
897	doi: 10.3389/fmicb.2019.00817	[Zika Virus Infection Induces Elevation of Tis...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
62	doi: 10.1002/path.5282	[PGC-1α deficiency causes spontaneous kidney i...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
350	doi: 10.1029/2018gl079826	[Understanding Rapid Adjustments to Diverse Fo...	[0, 2, 0, 0, 1, 0, 0]
677	doi: 10.1111/cts.12551	[Clinical and Functional Relevance of the Mono...	[0, 0, 0, 0]

Figure 5::dataset for argument II

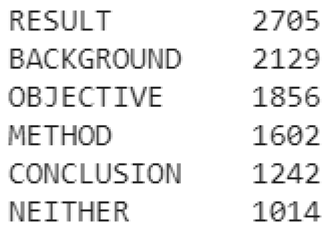
The dataset_aueb_structure_v2.json file contains the same information as the dataset_aueb_argument_v3.json counterpart. However, there is a great distinction in their annotation. The dataset_aueb_structure_v2.json file has labels with regards to a sentence's structure. These are Neither, Background, Objective, Method, Result, Conclusion. The dataset contains 1014 abstracts and the label distribution can be seen below.



```
Dataset length: 1014 abstracts
```

	document	sentences	labels
195	doi: 10.1016/j.future.2018.08.045	[Patterns for High Performance Multiscale Comp...	[NEITHER, BACKGROUND, BACKGROUND, BACKGROUND, ...
273	doi: 10.1016/j.wace.2019.100206	[Experiment design of the International CLIVAR...	[NEITHER, OBJECTIVE, METHOD, METHOD, METHOD, O...
956	doi: 10.3934/mine.2018.1.1	[A monolithic algorithm for the simulation of ...	[NEITHER, OBJECTIVE, METHOD, METHOD, METHOD, M...
68	doi: 10.1002/qj.3598	[Towards a more reliable historical reanalysis...	[NEITHER, OBJECTIVE, OBJECTIVE, BACKGROUND, BA...
47	doi: 10.1002/cphc.201800321	[Design of Perchlorotriphenylmethyl (PTM) Radi...	[NEITHER, BACKGROUND, BACKGROUND, BACKGROUND, ...

Figure 6::dataset for structure



```
RESULT        2705
BACKGROUND    2129
OBJECTIVE     1856
METHOD        1602
CONCLUSION    1242
NEITHER       1014
```

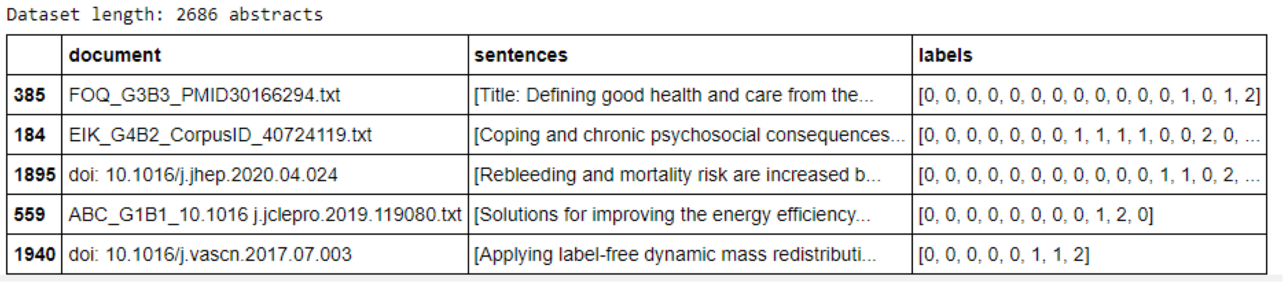
Figure 7: Label distribution in structure dataset

The dataset_aueb_citations_v2.json file contains 1 to 10 citations of an abstract to another. Moreover, additional information is included about the previous and the next sentence that the citation is from. Then the citations is labeled as positive, negative or neutral. Moreover, the file includes similar information as previous files about the abstract. The scope of this dataset is to be used for citation prediction. This dataset was not utilized.

The eu_calls file has information about a projects EU call. This dataset was not used in our analysis and could certainly be used as added feature to improve results of mining models.

2.4 Data Processing / Annotation

For the argument mining task the two different datasets (dataset.json and dataset_aueb_argument_v3.json) were combined to one dataset as they different abstracts. The resulting dataset contains 2686 abstracts or 32004 sentences. 22375 of these are labeled neither, 6210 are labeled evidence and 3419 are labeled claim.



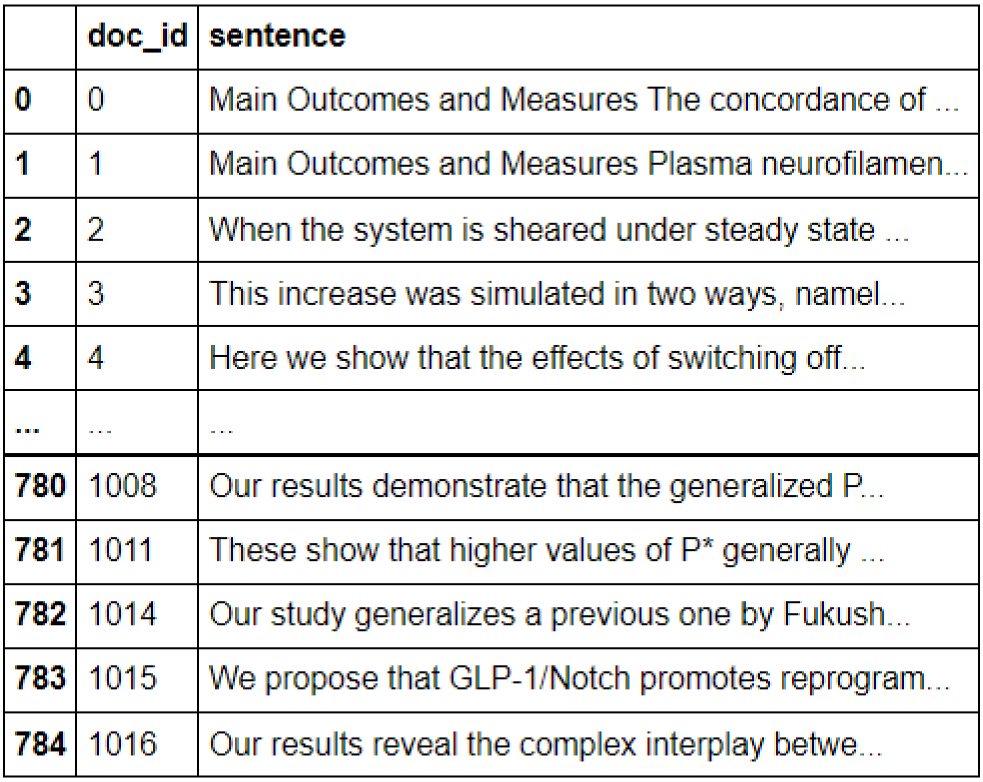
Dataset length: 2686 abstracts

	document	sentences	labels
385	FOQ_G3B3_PMID30166294.txt	[Title: Defining good health and care from the...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 2]
184	EIK_G4B2_CorpusID_40724119.txt	[Coping and chronic psychosocial consequences...	[0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 2, 0, ...
1895	doi: 10.1016/j.jhep.2020.04.024	[Rebleeding and mortality risk are increased b...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 2, ...
559	ABC_G1B1_10.1016 j.jclepro.2019.119080.txt	[Solutions for improving the energy efficiency...	[0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 0]
1940	doi: 10.1016/j.vascn.2017.07.003	[Applying label-free dynamic mass redistributi...	[0, 0, 0, 0, 0, 1, 1, 2]

Figure 8: dataset for argument combined

For the structure mining task, the existing dataset was used. However, we have had a problematic entry (the abstract with doi: 10.1111/jnc.13838), which was deleted manually from the dataset_aueb_structure_v2.json archive before being saved in the repository. This was the case as it contained irrelevant characters.

For clustering purposes after calculating sentence embeddings, the sentences only labeled as evidence or claim from the dataset_aueb_argument_v3.json file were used. These amounted to 2748 sentences of which 1700 were labeled as evidence and 1048 as claim and in total make up 785 abstracts. These were fed to the fasttext model to create the necessary sentence embeddings. Moreover, the SDG column was added to the original dataset so it could be used as attribute to the nodes.



	doc_id	sentence
0	0	Main Outcomes and Measures The concordance of ...
1	1	Main Outcomes and Measures Plasma neurofilamen...
2	2	When the system is sheared under steady state ...
3	3	This increase was simulated in two ways, namel...
4	4	Here we show that the effects of switching off...
...	...	...
780	1008	Our results demonstrate that the generalized P...
781	1011	These show that higher values of P* generally ...
782	1014	Our study generalizes a previous one by Fukush...
783	1015	We propose that GLP-1/Notch promotes reprogram...
784	1016	Our results reveal the complex interplay betwe...

Figure 9: Clustering dataset

To obtain better results to either the baseline models or the fasttext model data pre-processing was performed. However, after evaluating relevant metrics we concluded that this process will only be implemented for the baseline argument model (lexicon).

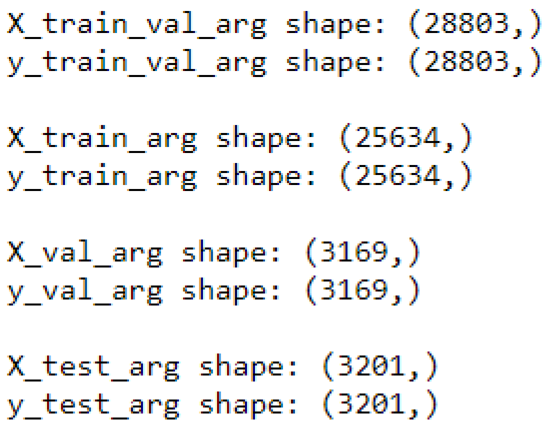
Specifically, for the vocabulary creation tokens were extracted from the sentences. These tokens included only alphabetical characters and stop words were removed. This was the case as stop words and symbols do not hold any relevant meaning or provide meaningful input to our model. What we call sentence tokenization is the procedure of splitting a sentence to its constructing words Moreover, all letters were transformed to lowercase. For this process the “nltk” library was used as this module provide built-in functions.

Inputs to are baseline models were the dataframes with the sentences of an abstract split to its rows.

For the fasttext models the split data were saved with the correct format for the model to .txt files so they can be used for later. This format is ‘__label__{0 or 1} sentence’.

To evaluate every model before input the data were split to train, validation and split.

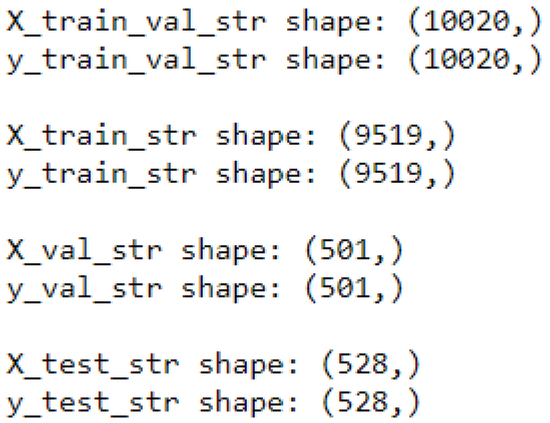
In more detail, for the argument mining task at first the data were split into two parts. The first one would be the Training-Validation dataset and the other would be the Test dataset. So, the Training-Validation dataset would include the 90% of the total 32004 sentences and Test dataset the 10% respectively. Then, we split again the Training-Validation dataset into Train (89 % of the train-validation set) and validation (11% of the train-validation set). Finally, we should clarify that we have used Stratified Random sampling. And we did that, because we needed to keep the proportion of occurrence of each label the same.



```
X_train_val_arg shape: (28803,)
y_train_val_arg shape: (28803,)

X_train_arg shape: (25634,)
y_train_arg shape: (25634,)

X_val_arg shape: (3169,)
y_val_arg shape: (3169,)

X_test_arg shape: (3201,)
y_test_arg shape: (3201,)
```

Figure 10: Train-validation split for arguments

The same concept was used for the implementation of the fasttext structure mining task where the data were split according to the bellow table.



```
X_train_val_str shape: (10020,)
y_train_val_str shape: (10020,)

X_train_str shape: (9519,)
y_train_str shape: (9519,)

X_val_str shape: (501,)
y_val_str shape: (501,)

X_test_str shape: (528,)
y_test_str shape: (528,)
```

Figure 11:: Train-validation split for structure

On the other hand, for the baseline structure mining with the label distribution approach, random sampling without replacement was performed. The data were split to train (90 % - abstracts) and test (10 % - 102 abstracts) and there was no validation set. The 102 abstracts contained 1017 sentences to be predicted.

Finally, the files were examined for duplicate DOI abstracts and none was found.

2.5.1 Baseline Algorithm for argument prediction

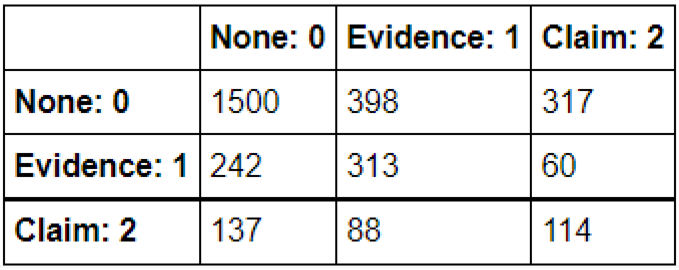
To begin with, our baseline model will be used for argumentation prediction purposes. Which means that we will try to build a model that predicts the label of a sentence (0 → None, 1 → Evidence, 2 → Claim). As a first step, we developed three lexicons which we created from the words in the training dataset. The first lexicon contains all the words from the training dataset, the second one only the words that are labeled as evidence (lexicon evidence), and the last one words that are labeled as claim (lexicon claim). From those lexicons, we extracted the 50 most common words and found the set difference between lexicon claim - lexicon all sentences. As a result, we found the top 50 words that are mostly contained in the sentences with a Claim label. In the same way, we worked to construct a second lexicon which contains the top 50 words that are mostly contained in the sentences with an Evidence label. The result of this process was a lexicon with 20 words that are mostly used in sentences labeled as Claim and a second lexicon with 22 words that are mostly used in sentences labeled as Evidence.

Another thing that we should take into consideration is that two words were common in the two lexicons (“increased”, “overall”). For that reason, we calculated how many times each word appears in each lexicon, and we decided that the words with the greater frequency in the lexicons will be characterized to the respective label. Which means that, in our case the word “increased” will be included in the evidence lexicon (the respective count was 242 versus 80) and the word “overall” will be included in the evidence lexicon (the respective count was 191 versus 87).

In the next step, we used the aforementioned lexicons in the Validation process. Three comparisons were performed. In particular, we used for validation:

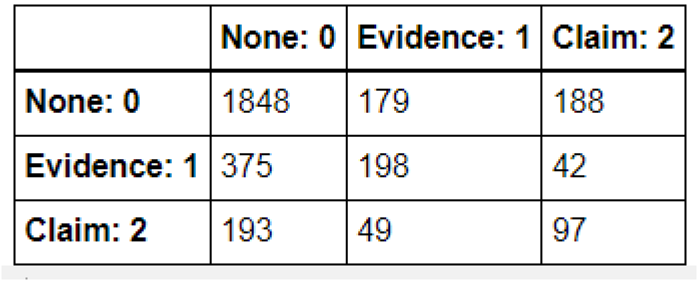
- All the words contained in the difference of the two sets
- The top 10 words contained in the difference of the two sets
- The top 5 words contained in the difference of the two sets

According to our results the best model was the one which used the 5 most common tokens. This can be seen in the tables below, where model III has the highest accuracy (71 %) between the three. This model is also used later to predict labels on the test dataset.



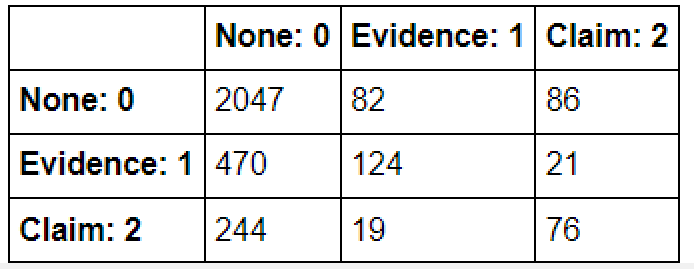
	None: 0	Evidence: 1	Claim: 2
None: 0	1500	398	317
Evidence: 1	242	313	60
Claim: 2	137	88	114

Figure 12: model all words baseline confusion matrix



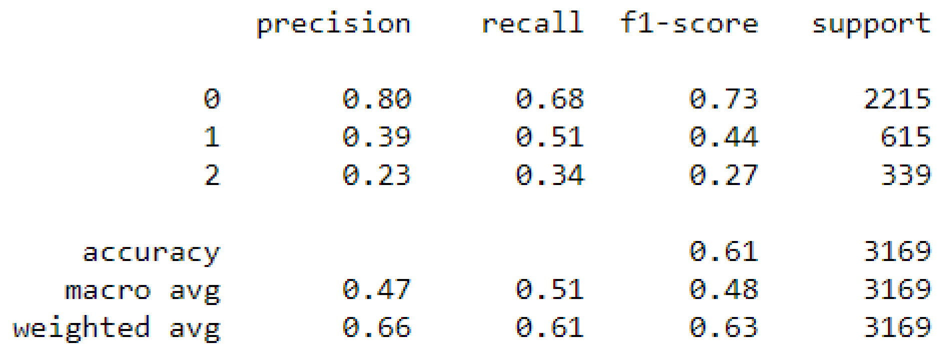
	None: 0	Evidence: 1	Claim: 2
None: 0	1848	179	188
Evidence: 1	375	198	42
Claim: 2	193	49	97

Figure 13: model top 10 words baseline confusion matrix



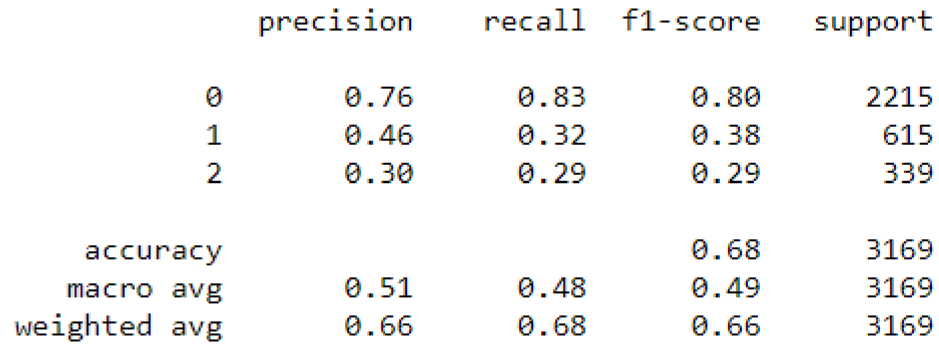
	None: 0	Evidence: 1	Claim: 2
None: 0	2047	82	86
Evidence: 1	470	124	21
Claim: 2	244	19	76

Figure 14: model top 5 words baseline confusion matrix



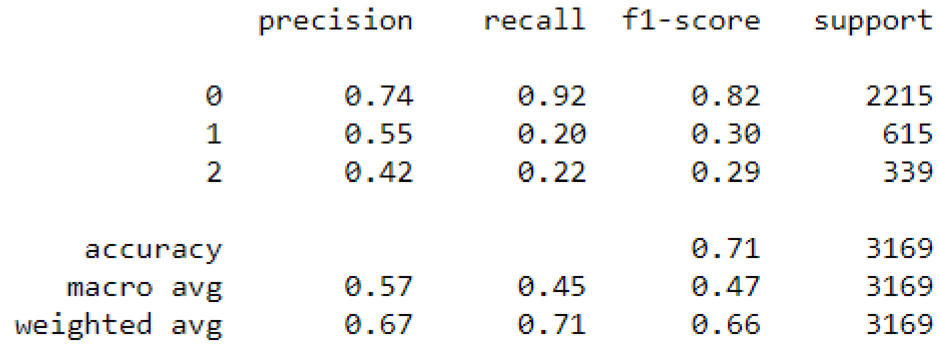
```
              precision    recall  f1-score   support

           0       0.80      0.68      0.73      2215
           1       0.39      0.51      0.44       615
           2       0.23      0.34      0.27       339

    accuracy                           0.61      3169
   macro avg       0.47      0.51      0.48      3169
weighted avg       0.66      0.61      0.63      3169
```

Figure 15: model all words baseline classification report



```
              precision    recall  f1-score   support

           0       0.76      0.83      0.80      2215
           1       0.46      0.32      0.38       615
           2       0.30      0.29      0.29       339

    accuracy                           0.68      3169
   macro avg       0.51      0.48      0.49      3169
weighted avg       0.66      0.68      0.66      3169
```

Figure 16: model top 10 words baseline classification report



```
              precision    recall  f1-score   support

           0       0.74      0.92      0.82      2215
           1       0.55      0.20      0.30       615
           2       0.42      0.22      0.29       339

    accuracy                           0.71      3169
   macro avg       0.57      0.45      0.47      3169
weighted avg       0.67      0.71      0.66      3169
```

Figure 17: model top 5 words baseline classification report

2.5.2 Baseline Algorithm for structure prediction

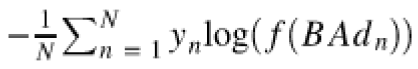
To begin with, our baseline model for Structure will be used for structure prediction purposes. Which means that we will try to build a model that for given number of sentences will give us as an answer for the label of the sentences (Neither/Background / Objective-Aim / Method / Result / Conclusion). So, in this approach the target is to create a dictionary from the training dataset that categorizes a sentence to a label based on its position. In more detail, in order to predict the label of a sentence our model compares this abstract with others that have the same number of sentences in them. Then, based on the sentence's position in the abstract, it selects the label that is most encountered in that position. If the abstract that is used for prediction has a number of sentences that have not been encountered in the training data set, then the abstract with the greatest number of sentences is used for prediction. This approach is based on the label's distribution in the abstracts.

2.5.3 Fasttext architecture

The model that we use for argument & structure prediction is named fasttext. The main principle behind fasttext is that the morphological structure of a word carries important information about the meaning of the word. Such structure is not considered by traditional word embeddings like Word2Vec, which train a unique word embedding for every individual word. This is especially significant for morphologically rich languages in which a single word can have a large number of morphological forms, each of which might occur rarely, thus making it hard to train good word embeddings. For that reason, fasttext attempts to solve this by treating each word as the aggregation of its sub words. For the sake of simplicity and language-independence, sub words are taken to be the character n-grams of the word. The vector for a word is simply taken to be the sum of all vectors of its component char-ngrams.

This improves accuracy of NLP related task, while maintaining speed. Furthermore, it provides context to the input similar to the way the RNN interprets the time series aspect and the CNN encodes the spatial aspect of the data.

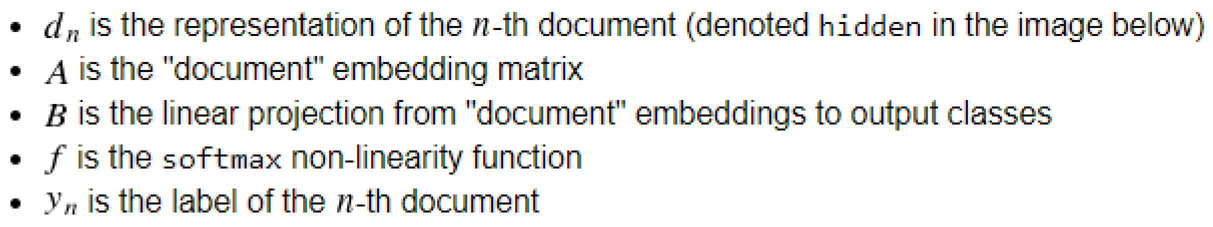
The actual model fasttext implements is rather simple as we can see in the image below -- the negative log-likelihood the model tries to minimize in training is



$$-\frac{1}{N}\sum_{n=1}^{N} y_n \log(f(BAd_n))$$

Figure 18: Fasttext Function

Where:



- d_n is the representation of the n-th document (denoted `hidden` in the image below)
- A is the "document" embedding matrix
- B is the linear projection from "document" embeddings to output classes
- f is the `softmax` non-linearity function
- y_n is the label of the n-th document

Figure 19: Fasttext function explanation

After, the n-grams have been created, the features are then averaged (pooling) and send to the hidden variables. Then we apply a SoftMax activation function to the output.

Once the input and hidden vectors are initialized, multiple training threads are kicked off. All the training threads hold a shared pointer to the matrices for input and hidden vectors. The threads all read from the input file, updating the model with each input line that is read, i.e stochastic gradient descent with a batch size of 1. An input line is truncated if newline character is encountered, or if the count of words we’ve read reaches the maximum allowed line size.

Both the continuous bag of words and the Skip-gram model update the weights for a context of size between a random uniform distribution between 1 and the value determined by the size of the context window.

The target vector for the loss function is computed via a normalized sum of all the input vectors. The input vectors are the vector representation for the original word, and all the n-grams of that word. The loss is computed which sets the weights for the forward pass, which propagate their way all the way back to the vectors for the input layer in the back propagation pass. This tuning of the input vector weights that happens during the back propagation pass is what allows us to learn representations that maximize co-occurrence similarity. The learning rate affects how much each instance affects the weights.

In our network, we have classification task with 3 classes (0 → None, 1 → Evidence, 2 → Claim). For that reason, we will use only 3 neurons in the output layer. We have also used activation function to squeeze the values into smaller range. The last layer activation function that we will be used is the SoftMax.

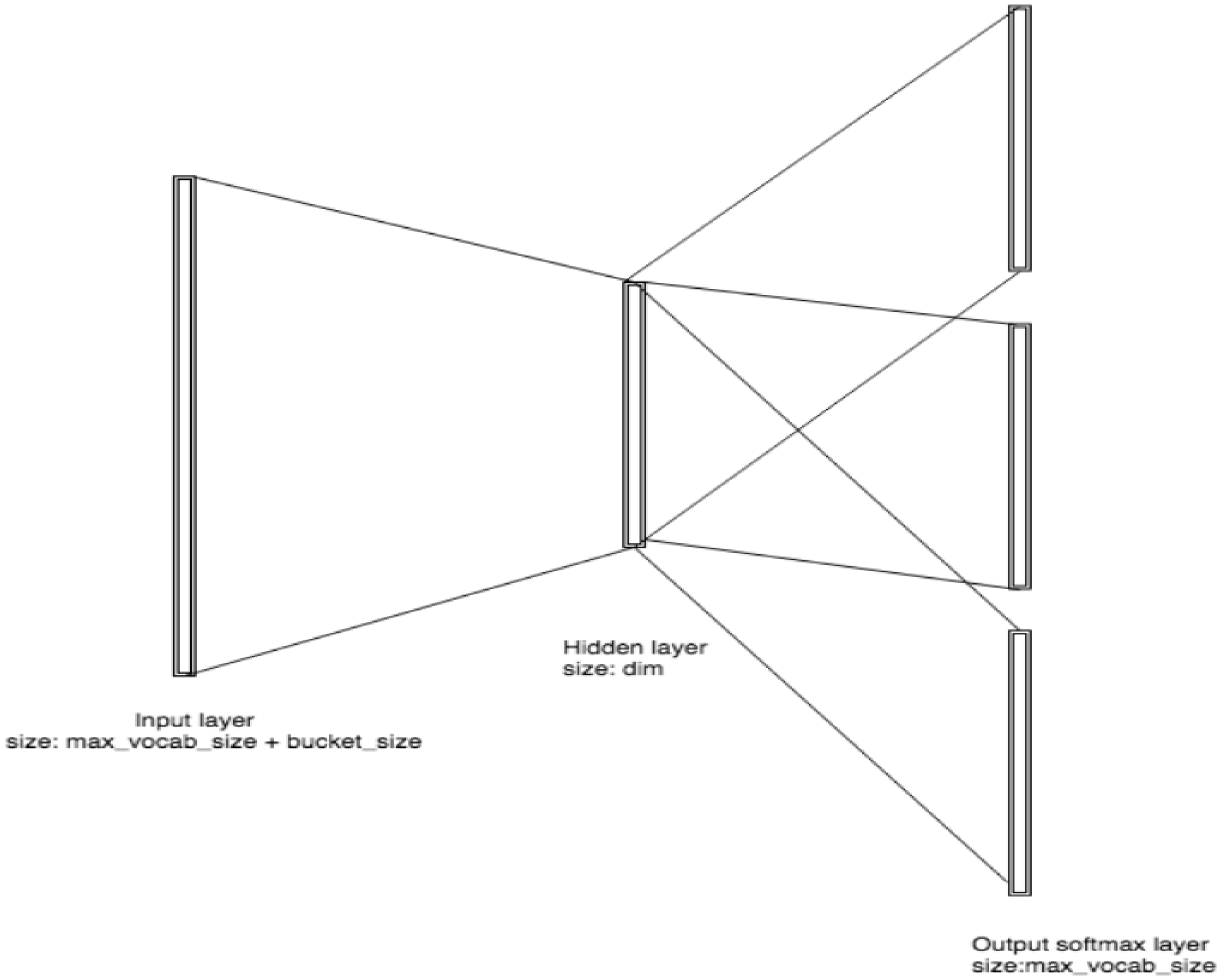


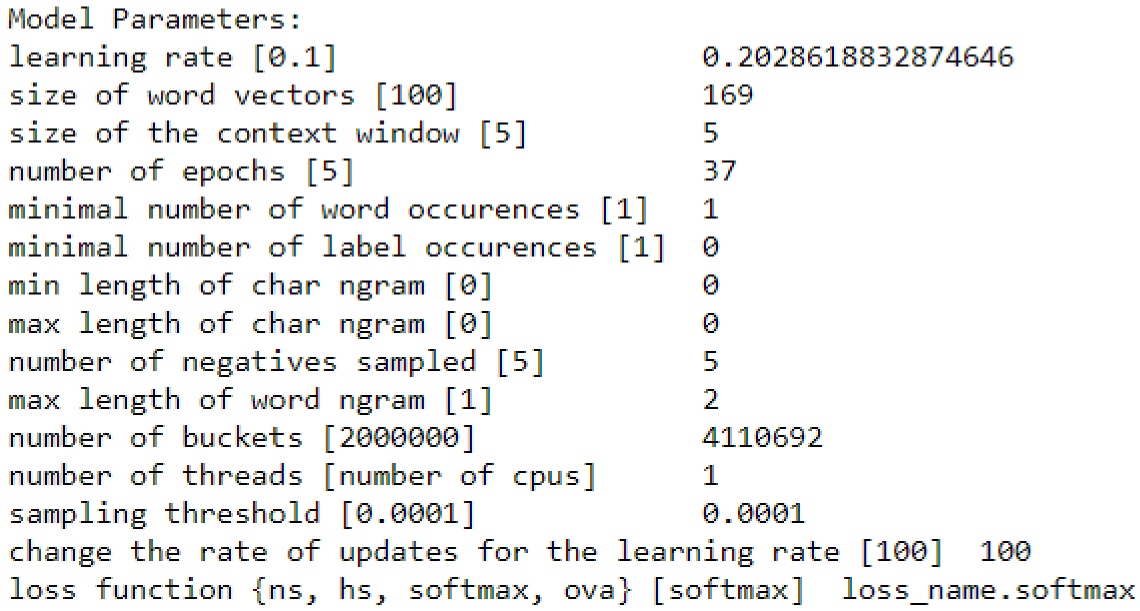
Figure 20: Fasttext model

The word embeddings calculated from the fasttext model applied in the arguments dataset are used to calculate sentence embeddings for sentences in abstracts that are labeled either evidence or claim. These sentence embeddings are used as inputs to measure cosine similarity between abstracts.

The aforementioned abstracts are then represented as nodes to a graph and connected to the 5 most similar abstracts that have as weight their similarity. Finally, cliques and communities are visualized and extracted from the graph according to Clauset-Newman-Moore greedy modularity maximization, Girvan-Newman Community Detection and clique percolation methods.

3.1.1 Fasttext configuration for Argument Mining

As previously mentioned, the original data were transformed to the appropriate inputs for the fasttext model. Then the model was executed for 5 minutes with automatic hyperparameter optimization on the validation dataset.



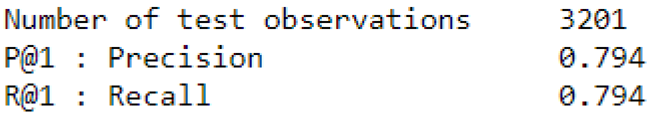
```
Model Parameters:
learning rate [0.1]                      0.2028618832874646
size of word vectors [100]               169
size of the context window [5]           5
number of epochs [5]                     37
minimal number of word occurences [1]    1
minimal number of label occurences [1]   0
min length of char ngram [0]             0
max length of char ngram [0]             0
number of negatives sampled [5]          5
max length of word ngram [1]             2
number of buckets [2000000]              4110692
number of threads [number of cpus]       1
sampling threshold [0.0001]              0.0001
change the rate of updates for the learning rate [100]  100
loss function {ns, hs, softmax, ova} [softmax]  loss_name.softmax
```

Figure 21: Fasttext argument hyperparameters

From the above table, we can extract information related to the parameters of the fast text model for Arguments. The model's learning rate is higher than the default value at 20 %, which may suggest along with other metrics signs of overfitting. The size of the word vector is 169 and is the size that is later used for our embeddings. The model managed to pass through 37 epochs in 5 minutes. Moreover, we did not use n-grams, but we had word N-grams of size 2. Finally, as the purpose of the training was multi-label classification the appropriate loss function was SoftMax.

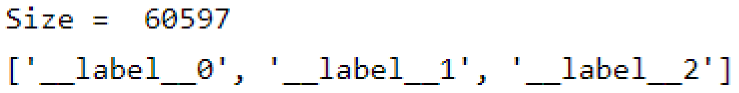
The results were evaluated with different pre-processing on the input data and the best recall and precision was achieved with no pre-processing at all.

In particular, the model achieved 79,4 % recall and 79,4 % precision on the test set, which correspond to the number of correct labels among the labels predicted by the model and the number of labels that were successfully predicted, among all the real labels accordingly. Moreover, the model managed to learn 60597 tokens for 3 labels.



```
Number of test observations     3201
P@1 : Precision                 0.794
R@1 : Recall                    0.794
```

Figure 22: Fasttext argument metrics



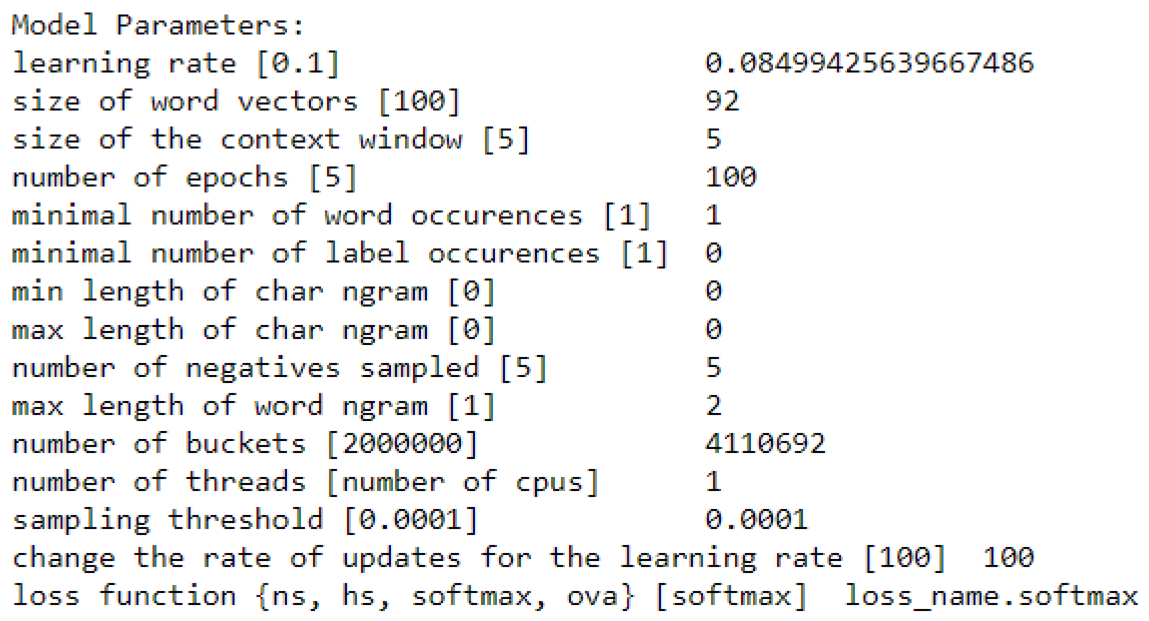
```
Size =  60597
['__label__0', '__label__1', '__label__2']
```

Figure 23: Fasttext argument size

Then by using the aforementioned inputs, word embeddings are used to create and visualize graph communities and cliques. The nodes represented in the graph are abstracts that have been labeled as evidence or claim. The union of these sentences in an abstract is parsed through our embeddings to get their representation. This representation is used to calculate the cosine similarity. The top 5 closest abstracts are connected to their counterparts in the graph with their weights. Communities are then explored through these methods Clauset-Newman-Moore greedy modularity maximization, Girvan-Newman Community Detection and clique percolation. Finally, the DOI abstract and the SDG are added as attributes to the graph.

3.1.2 Fasttext configuration for Structure Mining

Similar to argument mining the original data were transformed to the appropriate inputs for the fasttext model. Then the model was executed for 5 minutes with automatic hyperparameter optimization on the validation dataset.



```
Model Parameters:
learning rate [0.1]                        0.08499425639667486
size of word vectors [100]                 92
size of the context window [5]             5
number of epochs [5]                       100
minimal number of word occurences [1]      1
minimal number of label occurences [1]     0
min length of char ngram [0]               0
max length of char ngram [0]               0
number of negatives sampled [5]            5
max length of word ngram [1]               2
number of buckets [2000000]                4110692
number of threads [number of cpus]         1
sampling threshold [0.0001]                0.0001
change the rate of updates for the learning rate [100]  100
loss function {ns, hs, softmax, ova} [softmax]  loss_name.softmax
```

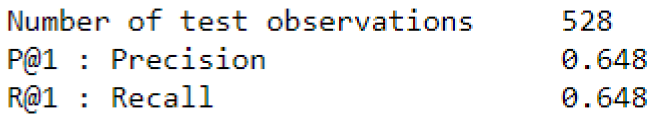
Figure 24: Fasttext structure hyperparameters

From the above table, we can extract information related to the parameters of the fast text model for Structure. The model’s learning rate is much lower than the default value at 20 %, The size of the word vector is 169 and is the size that is later used for our embeddings. The model managed to pass through 37 epochs in 5 minutes. Moreover, we did not use n-grams, but we had word N-grams of size 2. Finally, as the purpose of the training was multi-label classification the appropriate loss function was SoftMax.

From the above table, we can extract information related to the parameters of the fast text model for Arguments. What we can say is that the learning rate which represents the speed at which our model “learns” is 8.5% (which needs further improvement). Then, we can interpret the number of epochs. More or less, the number of epochs is a hyperparameter that defines the number of times that our algorithm will iterate through its input. In our model, that we have 100 epochs we can say that each an example has had plenty of opportunities (100) to update the internal model parameters. Moreover, the size of the words vectors in our model is 92 so less dimensions are needed to represent words. We did not use n-grams, but we had word N-grams of size 2. Finally, as the purpose of the training was multi-label classification the appropriate loss function was SoftMax.

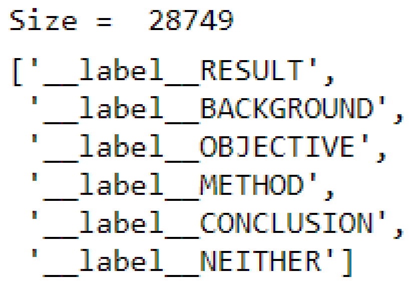
The results were evaluated with different pre-processing on the input data and the best recall and precision was achieved with lowercasing all alphabetical characters.

From the below table, we can discuss about two very useful metrics regarding to the evaluation of the fast text model. These are, the precision which is 64.8% and it represents the number of correct labels among the labels predicted by the model, and the Recall. The recall is again 64.8% and it represents the number of labels that successfully were predicted, among all the real labels. Finally, the model learned 28749 tokens for 6 labels.



```
Number of test observations     528
P@1 : Precision                 0.648
R@1 : Recall                    0.648
```

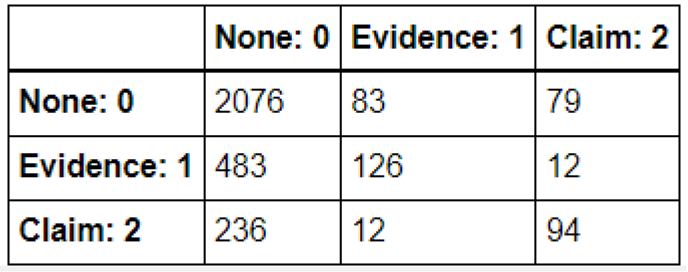
Figure 25: Fasttext structure metrics



```
Size =  28749
['__label__RESULT',
 '__label__BACKGROUND',
 '__label__OBJECTIVE',
 '__label__METHOD',
 '__label__CONCLUSION',
 '__label__NEITHER']
```

Figure 26: Fasttext structure size

3.2 Results & Quantitative analysis



	None: 0	**Evidence: 1**	**Claim: 2**
None: 0	2076	83	79
Evidence: 1	483	126	12
Claim: 2	236	12	94

Figure 27: Confusion Matrix baseline argument

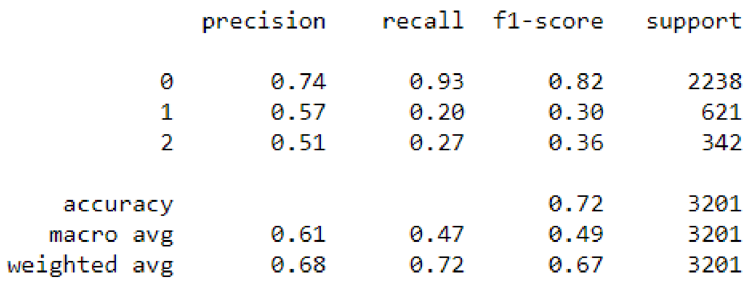
Related to the baseline model for arguments with the most 5 words we can say that:

- given that the sentence was labeled as None our model seems to correctly predict this label in most of the cases (92.7%).

- given that the sentence was labeled as Evidence our model seems to faulty predict that the label of this sentence is None (77.7%), and only in 20% of the cases this model correctly predict this label.

- given that the sentence was labeled as Claim our model seems to faulty predict that the label of this sentence is None (69%), and only in 27.5% of the cases this model correctly predict this label.

What we have mentioned above, tend us to think that the baseline model fails to realize most of the time if a sentence is labeled as evidence or claim. This can be confirmed by the classification report where the f1-score for the label None is 0.82. In contrast to the label evidence (0.30) and the label claim (0.36). Moreover, the Support metric which is related to the number of sentences that have an actual label and we can say that most of the sentences in the abstracts fall in the category of None (2238/3201= 69.9%).

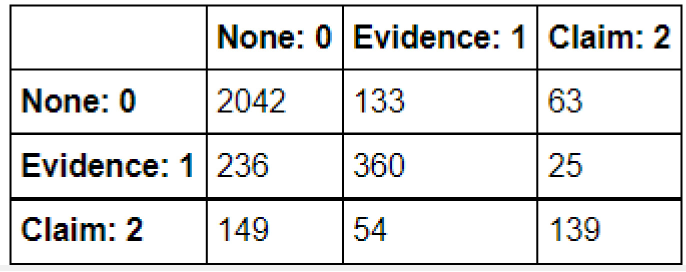


```
              precision    recall  f1-score   support

           0       0.74      0.93      0.82      2238
           1       0.57      0.20      0.30       621
           2       0.51      0.27      0.36       342

    accuracy                           0.72      3201
   macro avg       0.61      0.47      0.49      3201
weighted avg       0.68      0.72      0.67      3201
```

Figure 28: Classification report baseline argument

After taking into consideration the classification report for the baseline model for arguments we can say that:

- The accuracy metric is acceptable (72%)
- The precision value of the label (None), which is the percentage of predictive positives which were correctly classified were better (74%) than the precision values of the Evidence and Claim respectively.
- The recall value, which is the percentage of actual positives which were correctly classified is greater by far for the None label (93%) than the recall values of the Evidence and Claim respectively.
- The f1 score, which is the harmonic mean between them is greater by far again for the sentences with label None (82%), and not high enough for the rest of the other labeled sentences.

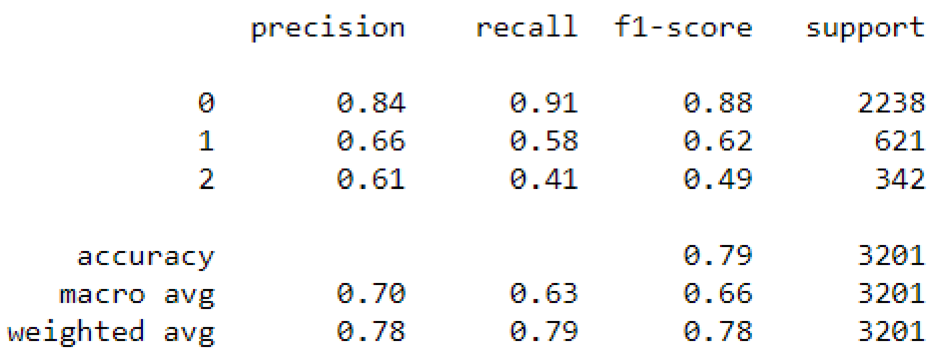


	None: 0	**Evidence: 1**	**Claim: 2**
None: 0	2042	133	63
Evidence: 1	236	360	25
Claim: 2	149	54	139

Figure 29: Confusion Matrix fasttext argument

Related to the fast text model for arguments we can say that:

- Given that the sentence was labeled as None our model seems to correctly predict this label in most of the cases (91.3%).
- Given that the sentence was labeled as Evidence our model seems to faulty predict that the label of this sentence is None (38%), and in 57.9% of the cases this model correctly predict this label.
- Given that the sentence was labeled as Claim our model seems to faulty predict that the label of this sentence is None (43.6%), and only in 40.7% of the cases this model correctly predict this label.



```
              precision    recall  f1-score   support

           0       0.84      0.91      0.88      2238
           1       0.66      0.58      0.62       621
           2       0.61      0.41      0.49       342

    accuracy                           0.79      3201
   macro avg       0.70      0.63      0.66      3201
weighted avg       0.78      0.79      0.78      3201
```

Figure 30: Classification report fasttext argument

After taking into consideration the Classification Report metrics so as to evaluate the fasttext model for arguments we can say that:

- The accuracy metric is acceptable (79%)

- The precision value of the label (None), which is the percentage of predictive positives which were correctly classified were better (84%) than the precision values of the Evidence and Claim respectively.

- The recall value, which is the percentage of actual positives which were correctly classified is greater by far for the None label (91%) than the recall values of the Evidence and Claim respectively.

- The f1 score, which is the harmonic mean between them is greater by far again for the sentences with label None (88%), and not high enough for the rest of the other labeled sentences.

Related to the comparison between the baseline model for arguments with the most 5 words and the fasttext model for arguments we can say that there are some agreements and some disagreements.

Agreements:

- Both models correctly predict that the label of the sentence is None in most of the cases.

- The recall value of the label None is by far greater than the rest of the other labels not only for the baseline model for arguments (93%) but also for the fasttext model for arguments (91%).

Disagreements:

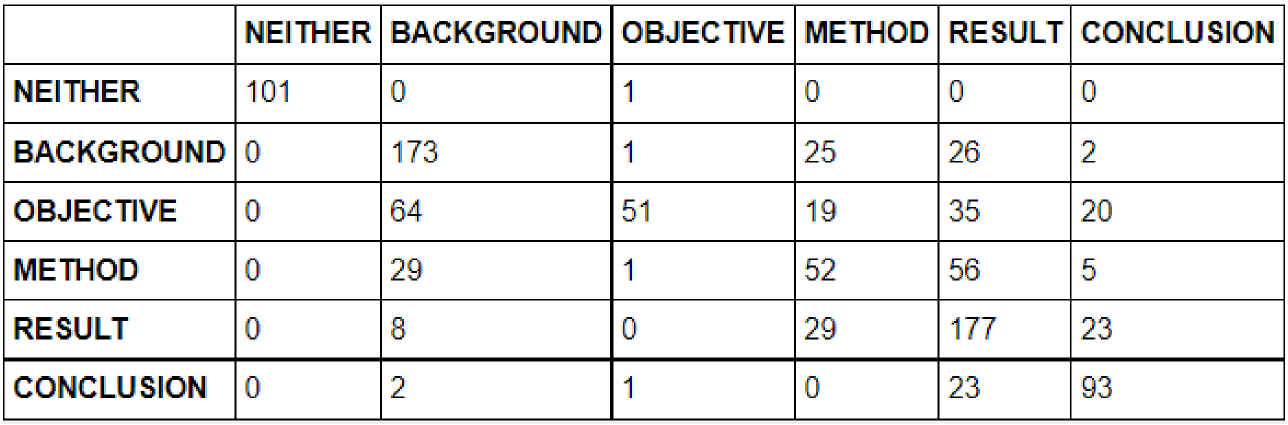
- Given that the sentence was labeled as Evidence our model seems to correctly predict the label of the sentence in 20% of the cases (baseline model for arguments) versus the fasttext model for arguments that correctly predict in 57.9%.

- Given that the sentence was labeled as Claim our model seems to correctly predict the label of the sentence in 27.5% of the cases (baseline model for arguments) versus the fasttext model for arguments that correctly predict in 40.7%.

- The accuracy metric is greater in the fasttext model (79%) in comparison with the baseline model for arguments (72%).

- The precision value of the label (None) is greater in the fasttext model for arguments (84%) in comparison with the baseline model for arguments (74%).

- The f1 score for the label None is greater in the fasttext model for arguments (88%) in comparison with the baseline model for arguments (82%).



	NEITHER	BACKGROUND	OBJECTIVE	METHOD	RESULT	CONCLUSION
NEITHER	101	0	1	0	0	0
BACKGROUND	0	173	1	25	26	2
OBJECTIVE	0	64	51	19	35	20
METHOD	0	29	1	52	56	5
RESULT	0	8	0	29	177	23
CONCLUSION	0	2	1	0	23	93

Figure 31: Confusion Matrix for baseline structure model

Related to the baseline model for structure with the most 5 words we can say that:

- Given that the sentence was labeled as Neither our model seems to correctly predict this label in most of the cases (99%).

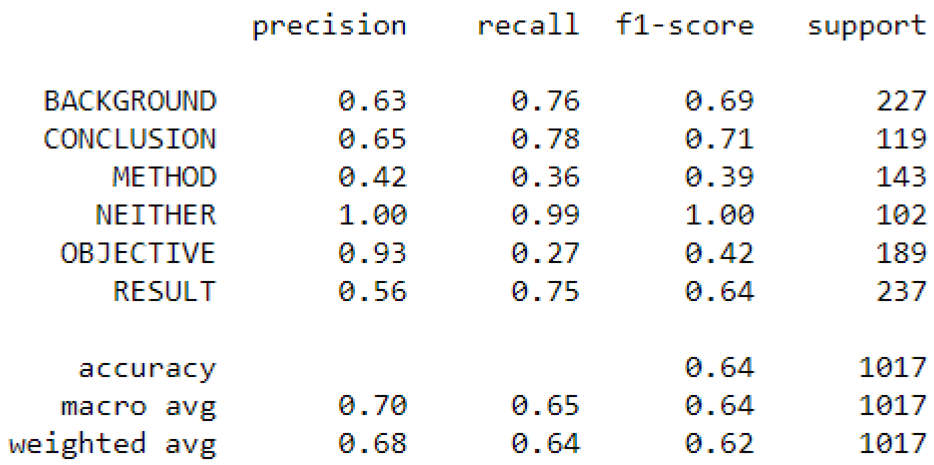
- Given that the sentence was labeled as Background our model seems to correctly predict that the label of this sentence is Background (76.2%), and if the label of the sentece is false then we tend to label them as Method (11%), or Result (11.45%) in most of the cases.

- Given that the sentence was labeled as Objective our model in most of the cases seems to faulty predict that the label of this sentence is Background (33.9%), and correctly predict that the label of this sentence is Objective (27%).

- Given that the sentence was labeled as Method our model in most of the cases seems to faulty predict that the label of this sentence is Result (39.2%), and correctly predict that the label of this sentence is Method (36.4%).

- Given that the sentence was labeled as Result our model seems to correctly predict that the label of this sentence is Result (74.7%), and if the label of the sentece is false then we tend to label them as Method (12.2%), or Conclusion (9.7%) in most of the cases.

- Given that the sentence was labeled as Conclusion our model seems to correctly predict that the label of this sentence is Conclusion (78.2%), and if the label of the sentece is false then we tend to label them as Result (19.3%) in most of the cases.

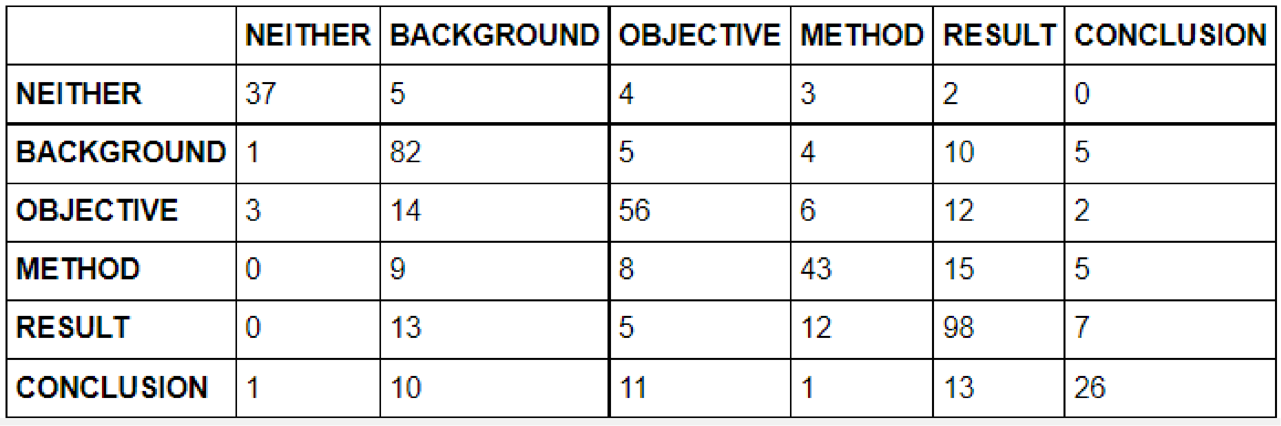


```
              precision    recall  f1-score   support

  BACKGROUND       0.63      0.76      0.69       227
  CONCLUSION       0.65      0.78      0.71       119
      METHOD       0.42      0.36      0.39       143
     NEITHER       1.00      0.99      1.00       102
   OBJECTIVE       0.93      0.27      0.42       189
      RESULT       0.56      0.75      0.64       237

    accuracy                           0.64      1017
   macro avg       0.70      0.65      0.64      1017
weighted avg       0.68      0.64      0.62      1017
```

Figure 32: Classification report for baseline structure model

After taking into consideration the Classification Report metrics so as to evaluate the baseline model for structure we can say that:

- The accuracy metric is acceptable (64%)
- The precision value of the classes, that is the percentage of predicted positives which were correctly classified were better for the labels Neither (100%), and Objective (93%) pretty good, and pretty much the same for the labels Background, Conclusion, Result, and Method.
- The value of the recall, that is the percentage of actual positives which were correctly classified is greater by far for the Neither label (99%), for the sentences with label Background, Conclusion, and Result pretty good, and for the rest of the labels not high enough.
- The f1-score, which is the harmonic mean between them is greater by far again for the sentences with label Neither (100%), pretty good values for the sentences with label Background, Conclusion, Result, and not high enough for the rest of the other labeled sentences.



	NEITHER	BACKGROUND	OBJECTIVE	METHOD	RESULT	CONCLUSION
NEITHER	37	5	4	3	2	0
BACKGROUND	1	82	5	4	10	5
OBJECTIVE	3	14	56	6	12	2
METHOD	0	9	8	43	15	5
RESULT	0	13	5	12	98	7
CONCLUSION	1	10	11	1	13	26

Figure 33: Confusion Matrix for fasttext structure model

Related to the fasttext model for structure we can say that:

- Given that the sentence was labeled as Neither our model seems to correctly predict this label in most of the cases (72.5%).

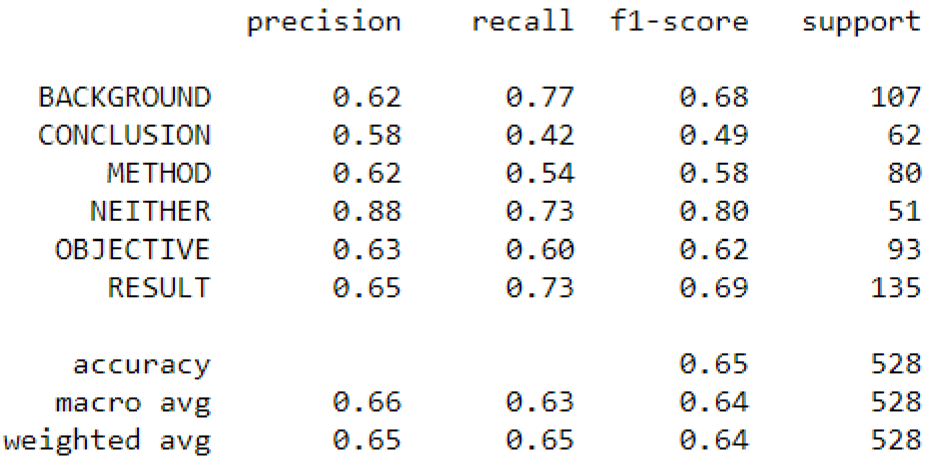
- Given that the sentence was labeled as Background our model seems to correctly predict that the label of this sentence is Background (76.6%), and if the label of the sentence is false then we tend to label them as Result in 10/107 times, and the rest of the other labels with no big difference at all.

- Given that the sentence was labeled as Objective our model seems to correctly predict that the label of this sentence is Objective (60%), and if the label of the sentence is false then we tend to label them as Background, and Result (most of the cases).

- Given that the sentence was labeled as Method our model in most of the cases seems to correctly predict that the label of this sentence is Method (53.8%), and if the label of the sentence is false then we tend to label them as Result (most of the cases).

- Given that the sentence was labeled as Result our model seems to correctly predict that the label of this sentence is Result (72.6%), and if the label of the sentence is false then we tend to label them as Background (%), or Method (%) in most of the cases.

- Given that the sentence was labeled as Conclusion our model seems to correctly predict that the label of this sentence is Conclusion (42%), and if the label of the sentence is false then we tend to label them as Result, Objective, or Background.



```
              precision    recall  f1-score   support

  BACKGROUND       0.62      0.77      0.68       107
  CONCLUSION       0.58      0.42      0.49        62
      METHOD       0.62      0.54      0.58        80
     NEITHER       0.88      0.73      0.80        51
   OBJECTIVE       0.63      0.60      0.62        93
      RESULT       0.65      0.73      0.69       135

    accuracy                           0.65       528
   macro avg       0.66      0.63      0.64       528
weighted avg       0.65      0.65      0.64       528
```

Figure 34:Classification report for fasttext structure model

After taking into consideration the Classification Report metrics so as to evaluate the baseline model for structure we can say that:

- The accuracy metric is acceptable (65%)

- The precision values of the classes which is the percentage of predicted positives which were correctly classified were better for the labels Neither (88%), and pretty much the same for the rest of the labels.

- The value of the recall, that is the percentage of actual positives which were correctly classified is greater for the Neither (73%), Result (73%), and Background label (77%), acceptable for the label Objective, and not high enough for the labels Method, and Conclusion.

- The f1-score, which is the harmonic mean between them is greater by far again for the sentences with label Neither (80%), pretty good values for the sentences with label

Background(68%), Objective (62%), Method (58%) and Result (69%), and not high enough for the sentences labeled as Conclusion.

Related to the comparison between the baseline model for structure with the most 5 words and the fasttext model for structure we can say that there are some agreements and some disagreements.

Agreements:

- Given that the sentence was labeled as Background our model seems to correctly predict the label of the sentence in 76.2% of the cases (baseline model for structure) versus the fasttext model for structure that correctly predict in 76.6%.

- Given that the sentence was labeled as Result our model seems to correctly predict the label of the sentence in 74.7% of the cases (baseline model for structure) versus the fasttext model for structure that correctly predict in 72.6%.

- The accuracy metric is pretty much the same between the fasttext model (65%), and the baseline model for structure (64%).

Disagreements:

- The baseline model for structure correctly predict that the label of the sentence is Neither (99%) greater than the fasttext model (72.5%).

- Given that the sentence was labeled as Objective our model seems to correctly predict the label of the sentence in 27% of the cases (baseline model for structure) versus the fasttext model for structure that correctly predict in 60%.

- Given that the sentence was labeled as Method our model seems to correctly predict the label of the sentence in 36.4% of the cases (baseline model for structure) versus the fasttext model for structure that correctly predict in 53.8%.

- Given that the sentence was labeled as Conclusion our model seems to correctly predict the label of the sentence in 78.2% of the cases (baseline model for structure) versus the fasttext model for structure that correctly predict in 42%.

- The precision value of the label (Neither) is greater in the baseline model for structure (100%) in comparison with the fasttext model for structure (88%).

- The recall value of the label Neither is greater in the baseline model for structure (99%) than in the fasttext model for structure (73%).

- The f1 score for the label Neither is greater in the baseline model for structure (100%) in comparison with the fasttext model for structure (80%).

The word embeddings calculated from our fasttext model for arguments were used to create clusters in graphs. The cosine similarity was used as a metric of weight between edges and nodes corresponded to abstracts. Edges were created only from and to abstracts that had embeddings that were in the top 5 closest neighbors.

In this graph we see the visualization of the connections between all the abstracts that contains sentences labeled as evidence or claim.

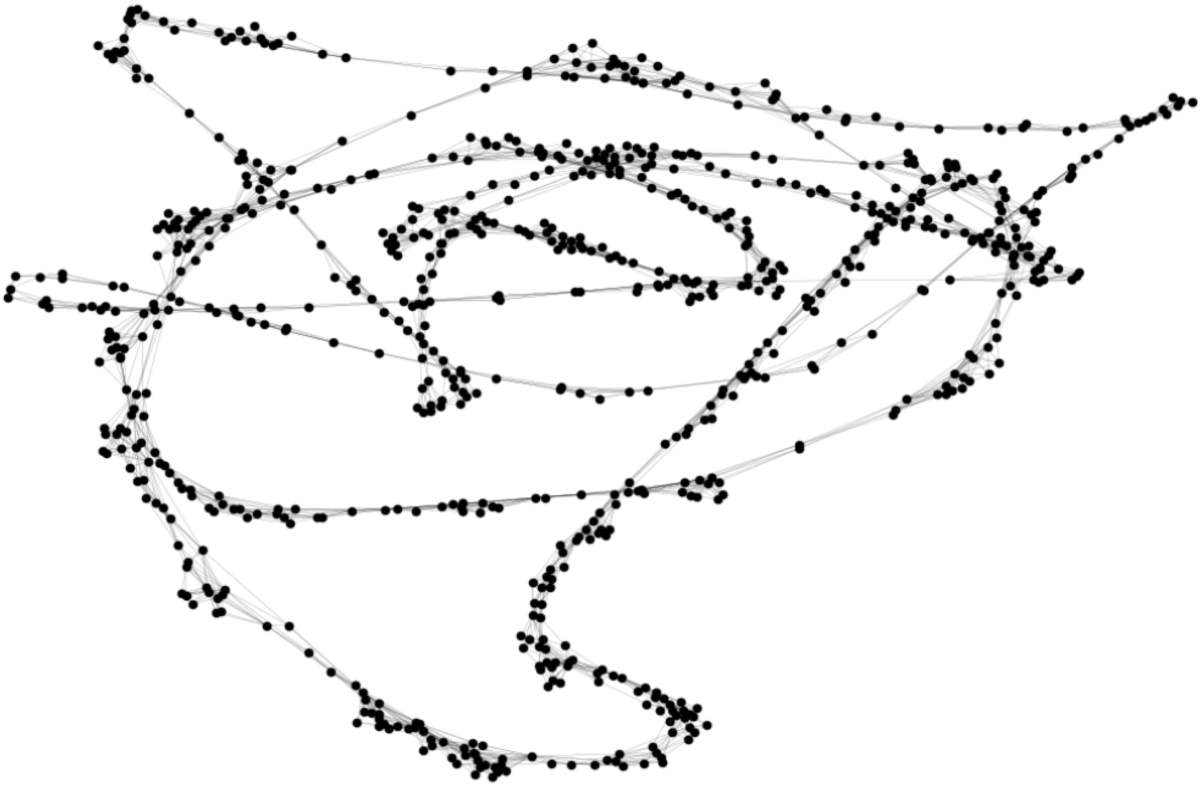


Figure 35:Graph

Moreover, according to Clauset – Newman - Moore with the greedy modularity maximization the graph contained 9 communities. A visualization of these communities can be seen below.

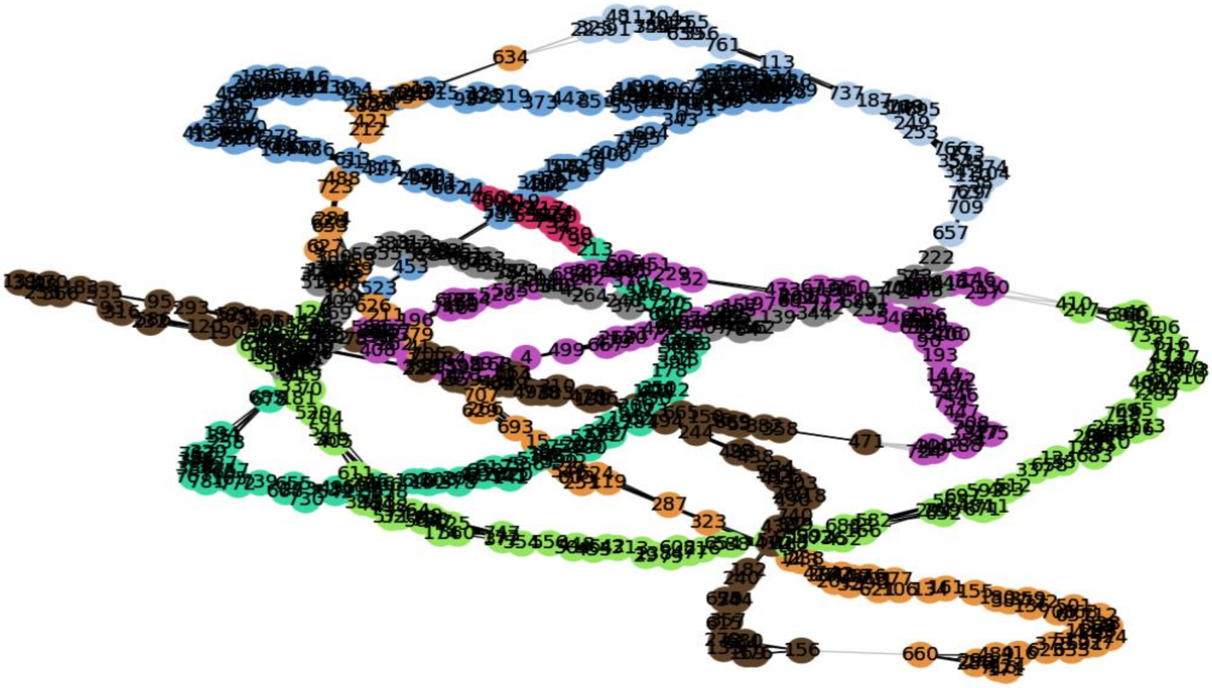


Figure 36: Clauset-Newman-Moore communities

On the other hand, according to Girvan-Newman for Community Detection the graph contained only 2 communities that can be seen below.

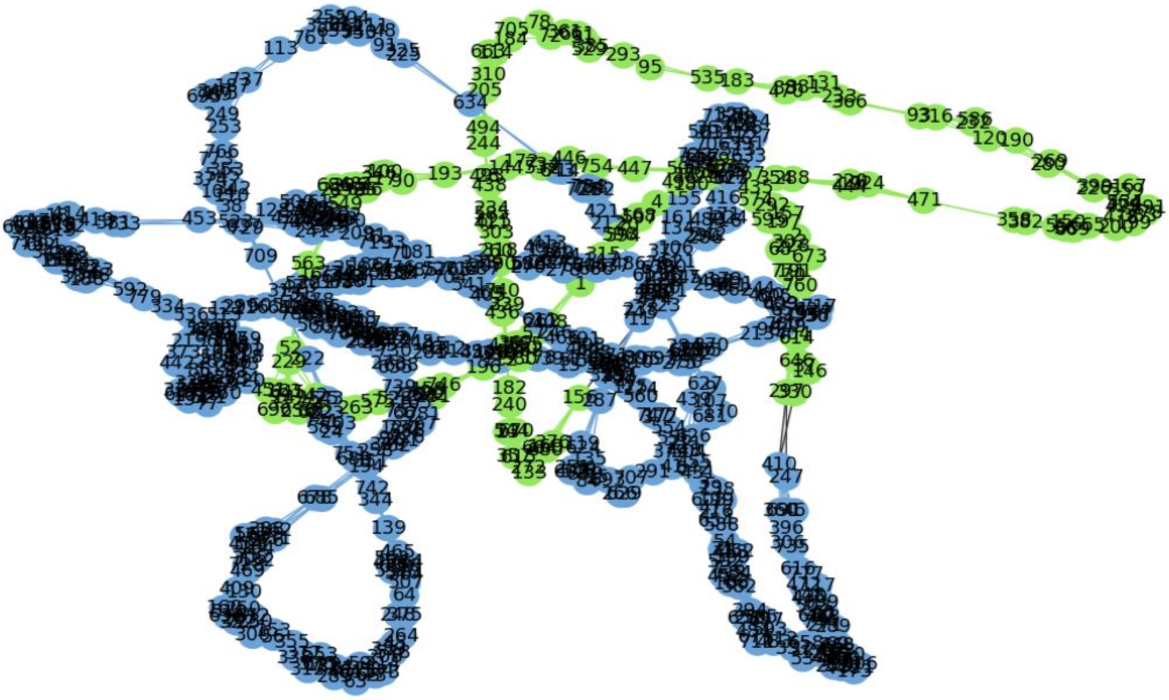


Figure 37:Girvan-Newman Community Detection

Lastly, the max number of nodes that a clique contained in our graph was 6. This along with node attributes and degree of each node can be seen with more detail be executing the code.

To visualize the average loss calculated per epoch we used a fasttext function from another module. This gave us different results and does not correspond to our actual model. However, the progress in reducing average loss per epoch can be seen hear.

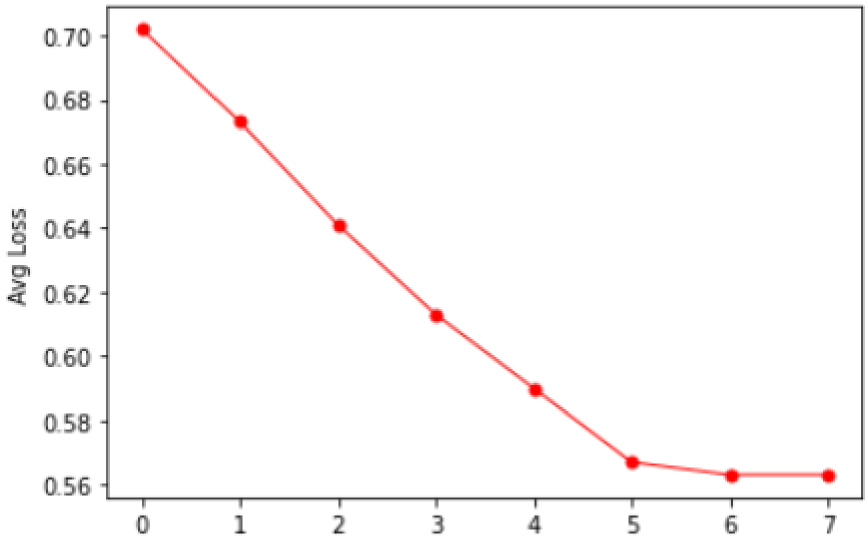


Figure 38: Avg loss plot fasttext argument

To visualize the average loss calculated per epoch we used a fasttext function from another module. This gave us different results and does not correspond to our actual model. However, the progress in reducing average loss per epoch can be seen hear.

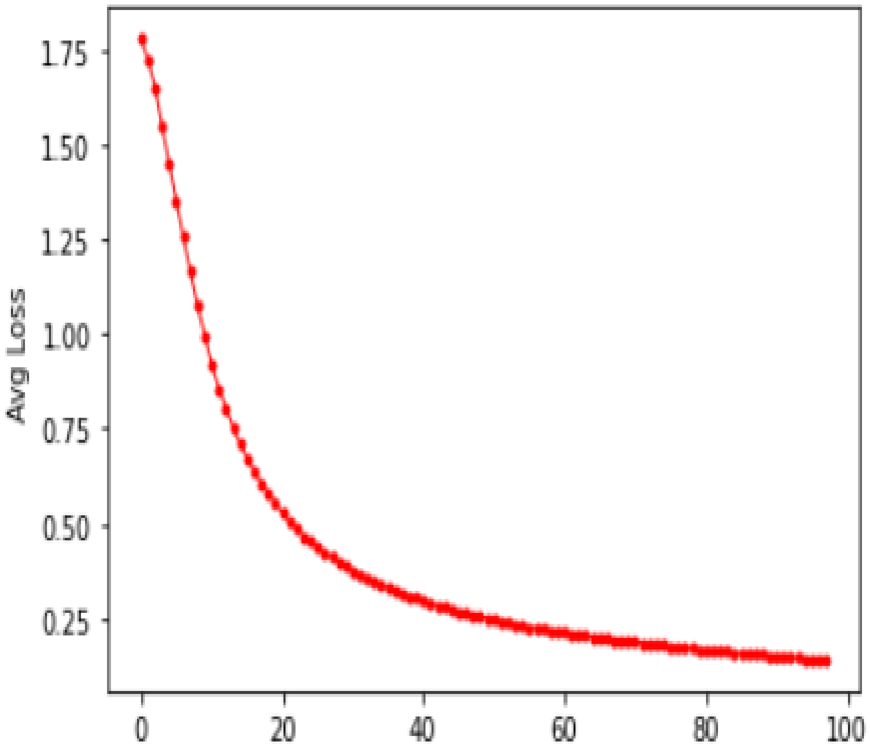


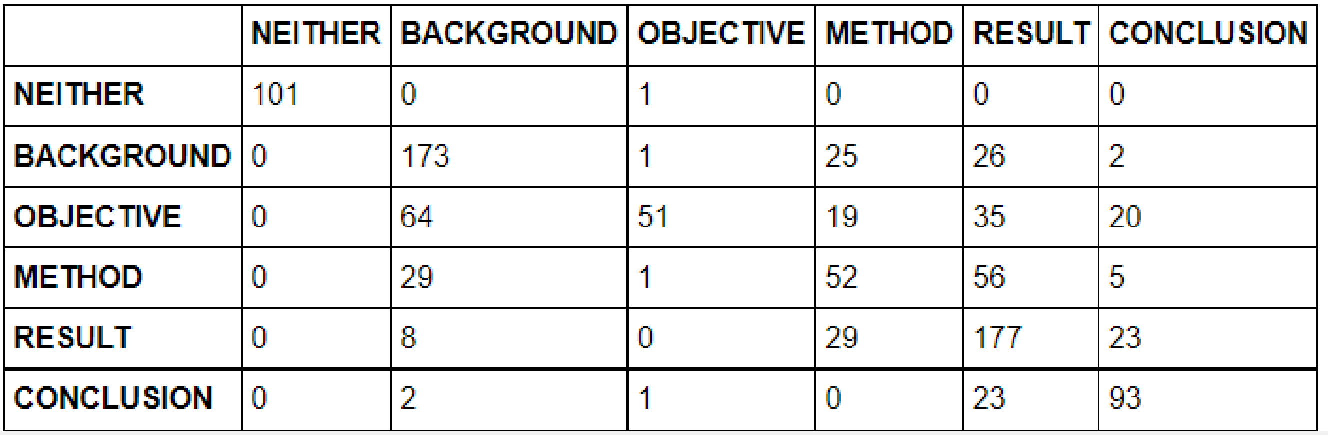
Figure 39: Avg loss plot fasttext structure

3.3 Qualitative & Error analysis

	Fasttext		
Model	TRAINING	VALIDATION	TESTING
ARGUMENT	0.99	0.78	0.79
STRUCTURE	0.99	0.64	0.65

Figure 40: Table of precision/recall for models

From this table, we can extract information related to the evaluation of our fast text models. More specifically, what is quite interesting is that in the training step our algorithm for both Argument and Structure prediction goes perfectly (99% of the sentences were predicted correctly from our models), but in the Testing and Validation step the results were not of the same quality. As we can see, in the Testing step the percentages were 79% for the Argument and 65% for the Structure. With that in mind we concluded that our fasttext model is overfitting to our training data. This is a huge issue, as our model fits exactly against its training data, and unfortunately cannot perform accurately against unseen data, defeating its purpose.



	NEITHER	BACKGROUND	OBJECTIVE	METHOD	RESULT	CONCLUSION
NEITHER	101	0	1	0	0	0
BACKGROUND	0	173	1	25	26	2
OBJECTIVE	0	64	51	19	35	20
METHOD	0	29	1	52	56	5
RESULT	0	8	0	29	177	23
CONCLUSION	0	2	1	0	23	93

Figure 41: Confusion matrix baseline structure

The above confusion matrix was used in the prediction of structure labels with our baseline model (label distribution method). From the results, we observe that 20 sentences that were labeled as objective by annotators, were predicted as conclusions by the model. The same case is seen for 35 wrongfully result predictions. As results and conclusions are usually the last sentences and this method uses the sentence's distribution to predict its label, this means that probably the model correctly predicted the sentence. In which case it could mean that some annotators did not make the correct annotation. This problem raises a more general issue on the quality of the data and the trustworthiness of the annotators. In the case of bad annotations even if the model has great predicted power, the predictions cannot be used in real life situations as the data do not correspond to the reality.

3.4 Future Work

To improve our predictions, it would be wise to use the word embeddings that we calculated as inputs to other models. This could include CNN, RNN or transformers. In addition, another option for our project was to extract context vectors from a pre-trained fast model and use these as input to an MLP.

Moreover, pre-processing techniques that were not applied such as lemmatization could improve the results of our models. This could also be coupled with data augmentation techniques for greater quantity and more diverse data.

Citance prediction is a topic that we could explore after implementing this basic approach.

Finally, there were information that was not used. This could had improved either argument mining or abstract clustering and provide more enriching results.

4.1 Members

The team consists of 2 members with similar background and studies. Moreover, all the members of the team contributed to the aliquot parts of the Pilot Project so as to deliver this final report. The processes needed included:

- Data Handling (Collection, Pre-Processing, Cleansing, and Preparation)
- Code development (in Python with the use of Google Collab)
- Building report (for interpretation purposes)

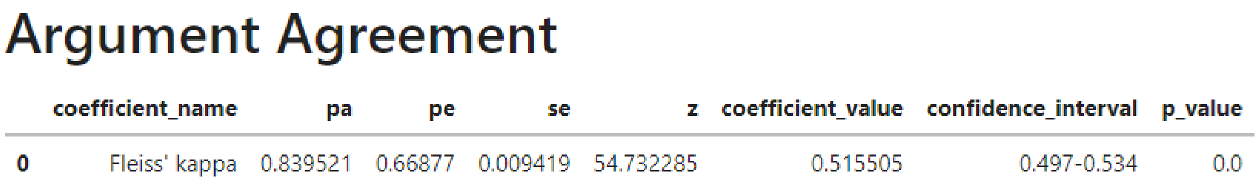
4.2 Time Plan

Regarding to the time management of the Pilot Project we can say that it in our initial plan we thought that it will take us about one and a half month (July 15 till 3 of September 3). More specifically, we had estimated that the step of how we would have taken the datasets it would have been by the end of July, and then we would have had the whole August so as to prepare the Data Handling, the Code Development, and the Building Report. Eventually, we needed more time so as to read the 125 abstracts carefully and label each sentence (Claim/Evidence/Background/Objective or Aim/Method/Result/Conclusion) and characterize each citance (positive/negative/neutral/irrelevant). For that reason, we had commenced the Data Handling and the rest of the processes in 17 of August which means that we had not plenty of time as we initially thought in the main tasks of this Project.

4.3 Bibliography

- https://fasttext.cc/docs/en/unsupervised-tutorial.html?fbclid=IwAR1MrsnU7qjloEHjF-DS94WtqxscslAHsfmv2Lw28zSoDClpCYOqvIxSo_M
- https://towardsdatascience.com/fasttext-under-the-hood-11efc57b2b3
- https://www.cis.lmu.de/esslli2017/pdf/print,embedgd.pdf?fbclid=IwAR3tbWWos_Yt0NTXZkef0dRWYC0I46254yXLq3NQhbQyywaOEw5eswkL3gY
- https://towardsdatascience.com/using-fasttext-and-svd-to-visualise-word-embeddings-instantly-5b8fa870c3d1
- https://orbifold.net/default/community-detection-using-networkx/
- https://networkx.org/documentation/stable/tutorial.html
- https://fasttext.cc/docs/en/python-module.html#important-preprocessing-data-encoding-conventions
- https://www.analyticsvidhya.com/blog/2020/04/community-detection-graphs-networks/
- https://www.cis.lmu.de/esslli2017/pdf/print,embedgd.pdf
- https://colab.research.google.com/github/NaiveNeuron/nlp-excercises/blob/master/tutorial2-fasttext/Text_Classification_fastText.ipynb#scrollTo=_DdFjtLj1qis
- Perakis G., Deep Learning Labs
- Papageorgiou H., Machine Learning & Content Analysis Slides

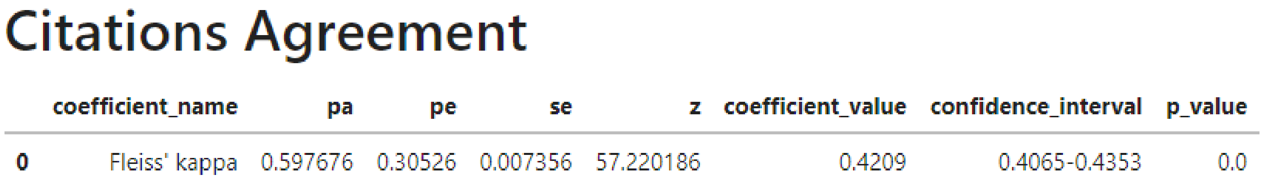
4.4 Appendices



Argument Agreement

	coefficient_name	pa	pe	se	z	coefficient_value	confidence_interval	p_value
0	Fleiss' kappa	0.839521	0.66877	0.009419	54.732285	0.515505	0.497-0.534	0.0

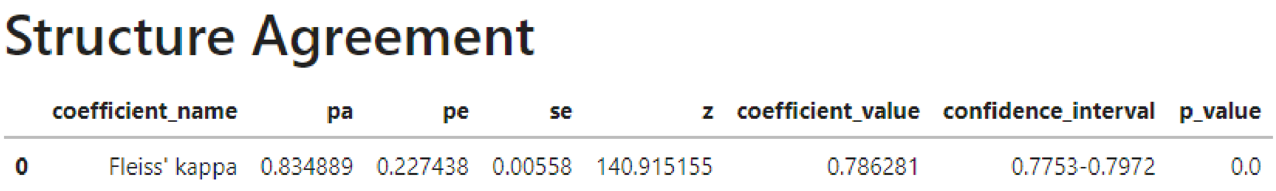
Figure 42: Argument Agreement



Citations Agreement

	coefficient_name	pa	pe	se	z	coefficient_value	confidence_interval	p_value
0	Fleiss' kappa	0.597676	0.30526	0.007356	57.220186	0.4209	0.4065-0.4353	0.0

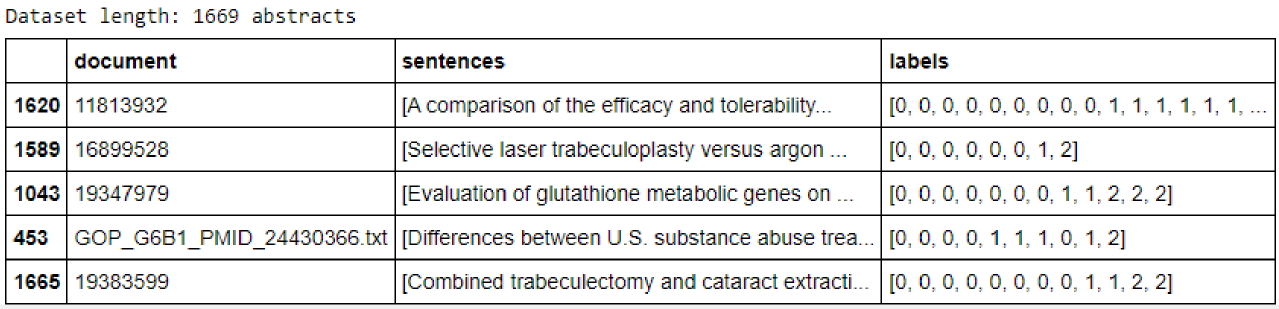
Figure 43: Citations Agreement



Structure Agreement

	coefficient_name	pa	pe	se	z	coefficient_value	confidence_interval	p_value
0	Fleiss' kappa	0.834889	0.227438	0.00558	140.915155	0.786281	0.7753-0.7972	0.0

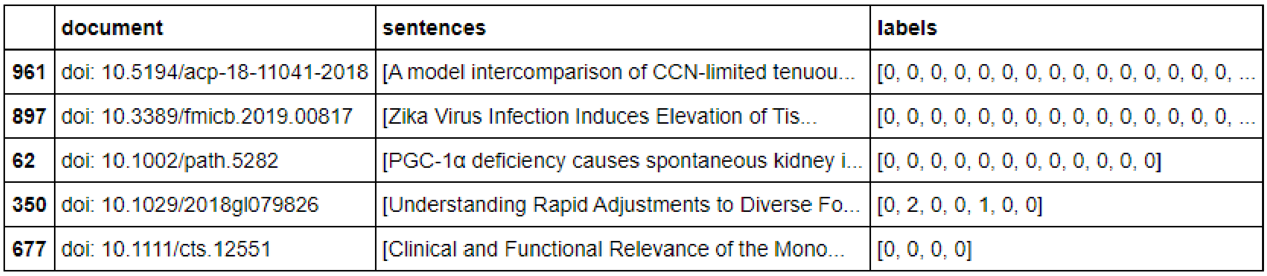
Figure 44: Structure Agreement



```
Dataset length: 1669 abstracts
```

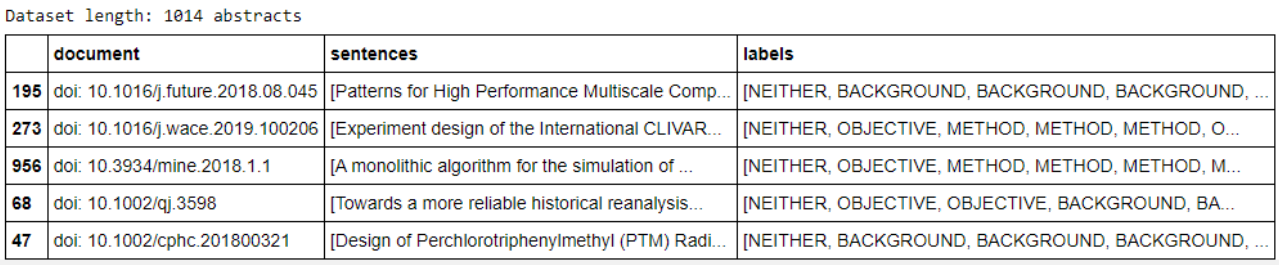
	document	sentences	labels
1620	11813932	[A comparison of the efficacy and tolerability...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, ...
1589	16899528	[Selective laser trabeculoplasty versus argon ...	[0, 0, 0, 0, 0, 0, 1, 2]
1043	19347979	[Evaluation of glutathione metabolic genes on ...	[0, 0, 0, 0, 0, 0, 0, 1, 1, 2, 2, 2]
453	GOP_G6B1_PMID_24430366.txt	[Differences between U.S. substance abuse trea...	[0, 0, 0, 0, 1, 1, 1, 0, 1, 2]
1665	19383599	[Combined trabeculectomy and cataract extracti...	[0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 2, 2]

Figure 45:dataset for argument I



	document	sentences	labels
961	doi: 10.5194/acp-18-11041-2018	[A model intercomparison of CCN-limited tenuou...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
897	doi: 10.3389/fmicb.2019.00817	[Zika Virus Infection Induces Elevation of Tis...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
62	doi: 10.1002/path.5282	[PGC-1α deficiency causes spontaneous kidney i...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
350	doi: 10.1029/2018gl079826	[Understanding Rapid Adjustments to Diverse Fo...	[0, 2, 0, 0, 1, 0, 0]
677	doi: 10.1111/cts.12551	[Clinical and Functional Relevance of the Mono...	[0, 0, 0, 0]

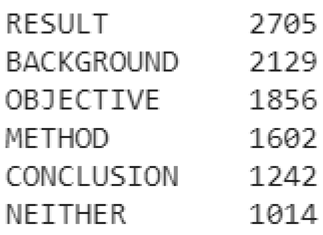
Figure 46::dataset for argument II



```
Dataset length: 1014 abstracts
```

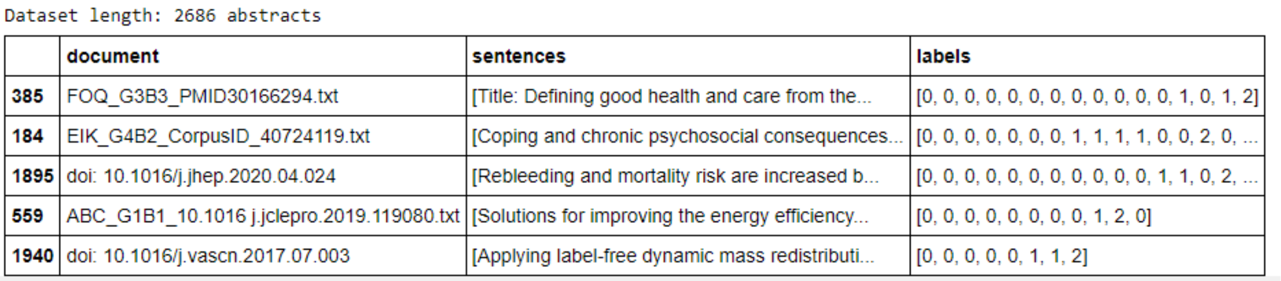
	document	sentences	labels
195	doi: 10.1016/j.future.2018.08.045	[Patterns for High Performance Multiscale Comp...	[NEITHER, BACKGROUND, BACKGROUND, BACKGROUND, ...
273	doi: 10.1016/j.wace.2019.100206	[Experiment design of the International CLIVAR...	[NEITHER, OBJECTIVE, METHOD, METHOD, METHOD, O...
956	doi: 10.3934/mine.2018.1.1	[A monolithic algorithm for the simulation of ...	[NEITHER, OBJECTIVE, METHOD, METHOD, METHOD, M...
68	doi: 10.1002/qj.3598	[Towards a more reliable historical reanalysis...	[NEITHER, OBJECTIVE, OBJECTIVE, BACKGROUND, BA...
47	doi: 10.1002/cphc.201800321	[Design of Perchlorotriphenylmethyl (PTM) Radi...	[NEITHER, BACKGROUND, BACKGROUND, BACKGROUND, ...

Figure 47::dataset for structure



```
RESULT        2705
BACKGROUND    2129
OBJECTIVE     1856
METHOD        1602
CONCLUSION    1242
NEITHER       1014
```

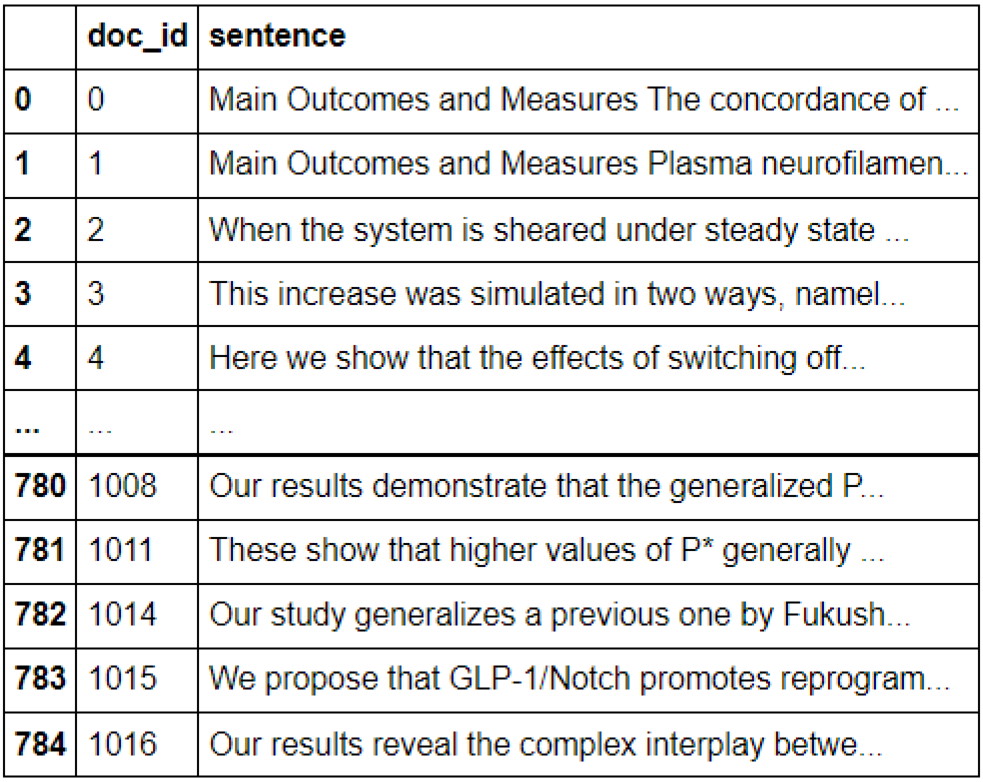
Figure 48: Label distribution in structure dataset



```
Dataset length: 2686 abstracts
```

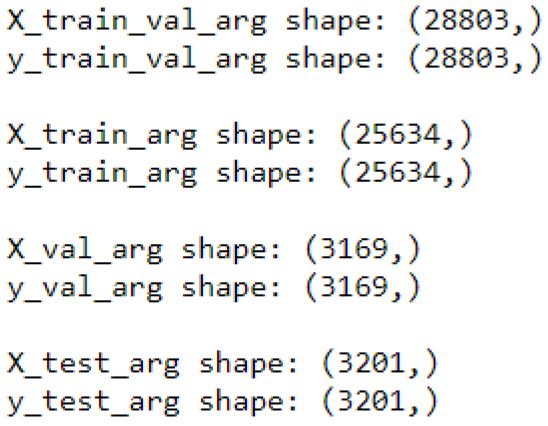
	document	sentences	labels
385	FOQ_G3B3_PMID30166294.txt	[Title: Defining good health and care from the...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 2]
184	EIK_G4B2_CorpusID_40724119.txt	[Coping and chronic psychosocial consequences...	[0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 2, 0, ...
1895	doi: 10.1016/j.jhep.2020.04.024	[Rebleeding and mortality risk are increased b...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 2, ...
559	ABC_G1B1_10.1016 j.jclepro.2019.119080.txt	[Solutions for improving the energy efficiency...	[0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 0]
1940	doi: 10.1016/j.vascn.2017.07.003	[Applying label-free dynamic mass redistributi...	[0, 0, 0, 0, 0, 1, 1, 2]

Figure 49: dataset for argument combined



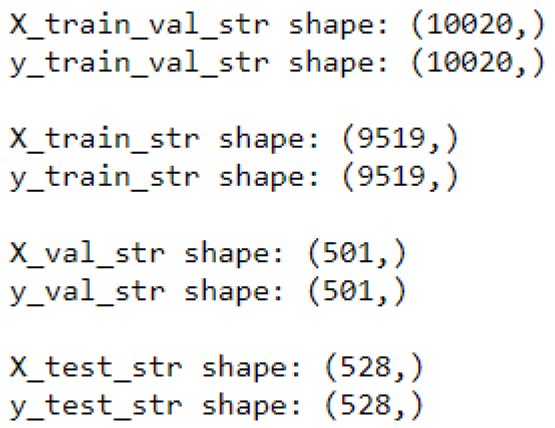
	doc_id	sentence
0	0	Main Outcomes and Measures The concordance of ...
1	1	Main Outcomes and Measures Plasma neurofilamen...
2	2	When the system is sheared under steady state ...
3	3	This increase was simulated in two ways, namel...
4	4	Here we show that the effects of switching off...
...	...	...
780	1008	Our results demonstrate that the generalized P...
781	1011	These show that higher values of P* generally ...
782	1014	Our study generalizes a previous one by Fukush...
783	1015	We propose that GLP-1/Notch promotes reprogram...
784	1016	Our results reveal the complex interplay betwe...

Figure 50: Clustering dataset



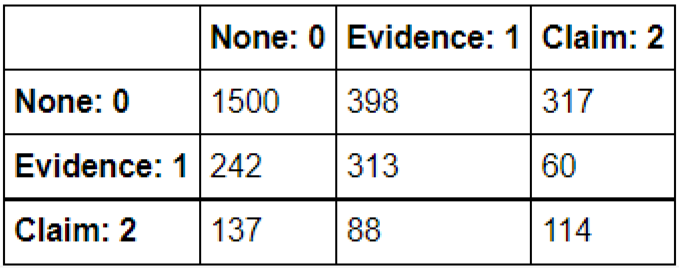
```
X_train_val_arg shape: (28803,)
y_train_val_arg shape: (28803,)

X_train_arg shape: (25634,)
y_train_arg shape: (25634,)

X_val_arg shape: (3169,)
y_val_arg shape: (3169,)

X_test_arg shape: (3201,)
y_test_arg shape: (3201,)
```

Figure 51: Train-validation split for arguments



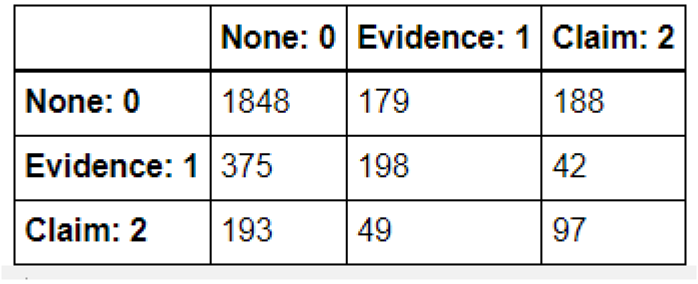
```
X_train_val_str shape: (10020,)
y_train_val_str shape: (10020,)

X_train_str shape: (9519,)
y_train_str shape: (9519,)

X_val_str shape: (501,)
y_val_str shape: (501,)

X_test_str shape: (528,)
y_test_str shape: (528,)
```

Figure 52:: Train-validation split for structure



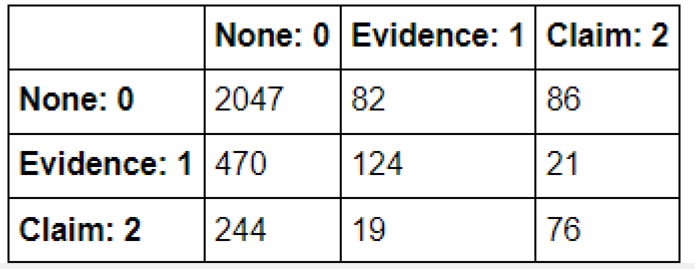
	None: 0	Evidence: 1	Claim: 2
None: 0	1500	398	317
Evidence: 1	242	313	60
Claim: 2	137	88	114

Figure 53: model all words baseline confusion matrix



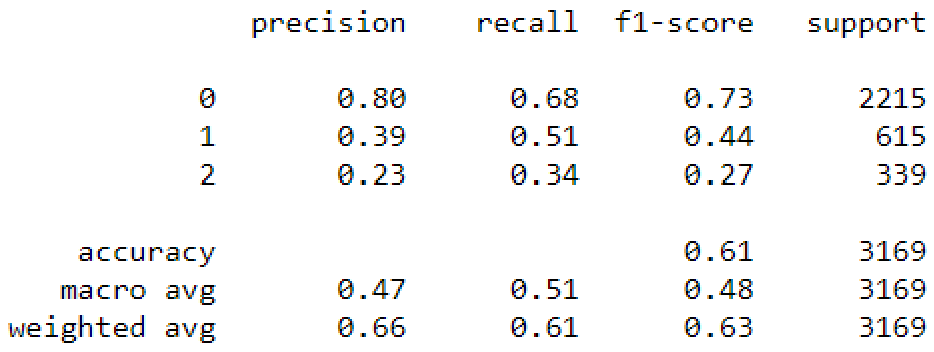
	None: 0	Evidence: 1	Claim: 2
None: 0	1848	179	188
Evidence: 1	375	198	42
Claim: 2	193	49	97

Figure 54: model top 10 words baseline confusion matrix



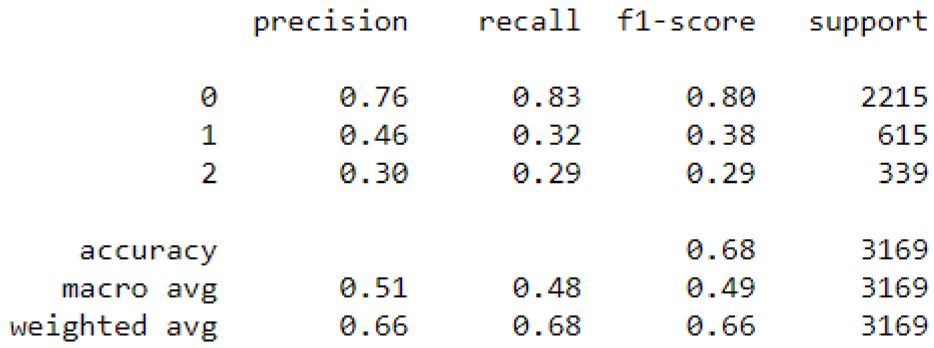
	None: 0	Evidence: 1	Claim: 2
None: 0	2047	82	86
Evidence: 1	470	124	21
Claim: 2	244	19	76

Figure 55: model top 5 words baseline confusion matrix



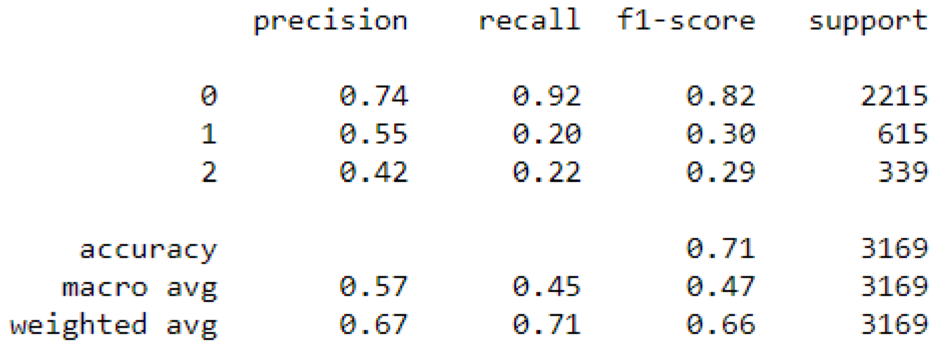
```
              precision    recall  f1-score   support

           0       0.80      0.68      0.73      2215
           1       0.39      0.51      0.44       615
           2       0.23      0.34      0.27       339

    accuracy                           0.61      3169
   macro avg       0.47      0.51      0.48      3169
weighted avg       0.66      0.61      0.63      3169
```

Figure 56: model all words baseline classification report



```
              precision    recall  f1-score   support

           0       0.76      0.83      0.80      2215
           1       0.46      0.32      0.38       615
           2       0.30      0.29      0.29       339

    accuracy                           0.68      3169
   macro avg       0.51      0.48      0.49      3169
weighted avg       0.66      0.68      0.66      3169
```

Figure 57: model top 10 words baseline classification report

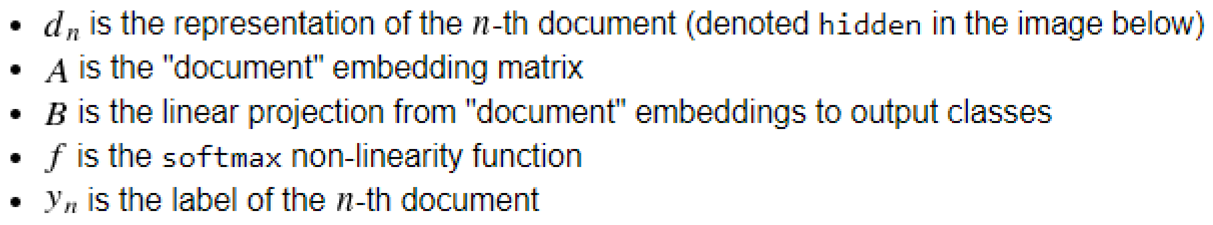


```
              precision    recall  f1-score   support

           0       0.74      0.92      0.82      2215
           1       0.55      0.20      0.30       615
           2       0.42      0.22      0.29       339

    accuracy                           0.71      3169
   macro avg       0.57      0.45      0.47      3169
weighted avg       0.67      0.71      0.66      3169
```

Figure 58: model top 5 words baseline classification report

$$-\frac{1}{N}\sum_{n=1}^{N} y_n \log(f(BAd_n))$$

Figure 59: Fasttext Function



- d_n is the representation of the n-th document (denoted `hidden` in the image below)
- A is the "document" embedding matrix
- B is the linear projection from "document" embeddings to output classes
- f is the `softmax` non-linearity function
- y_n is the label of the n-th document

Figure 60: Fasttext function explanation

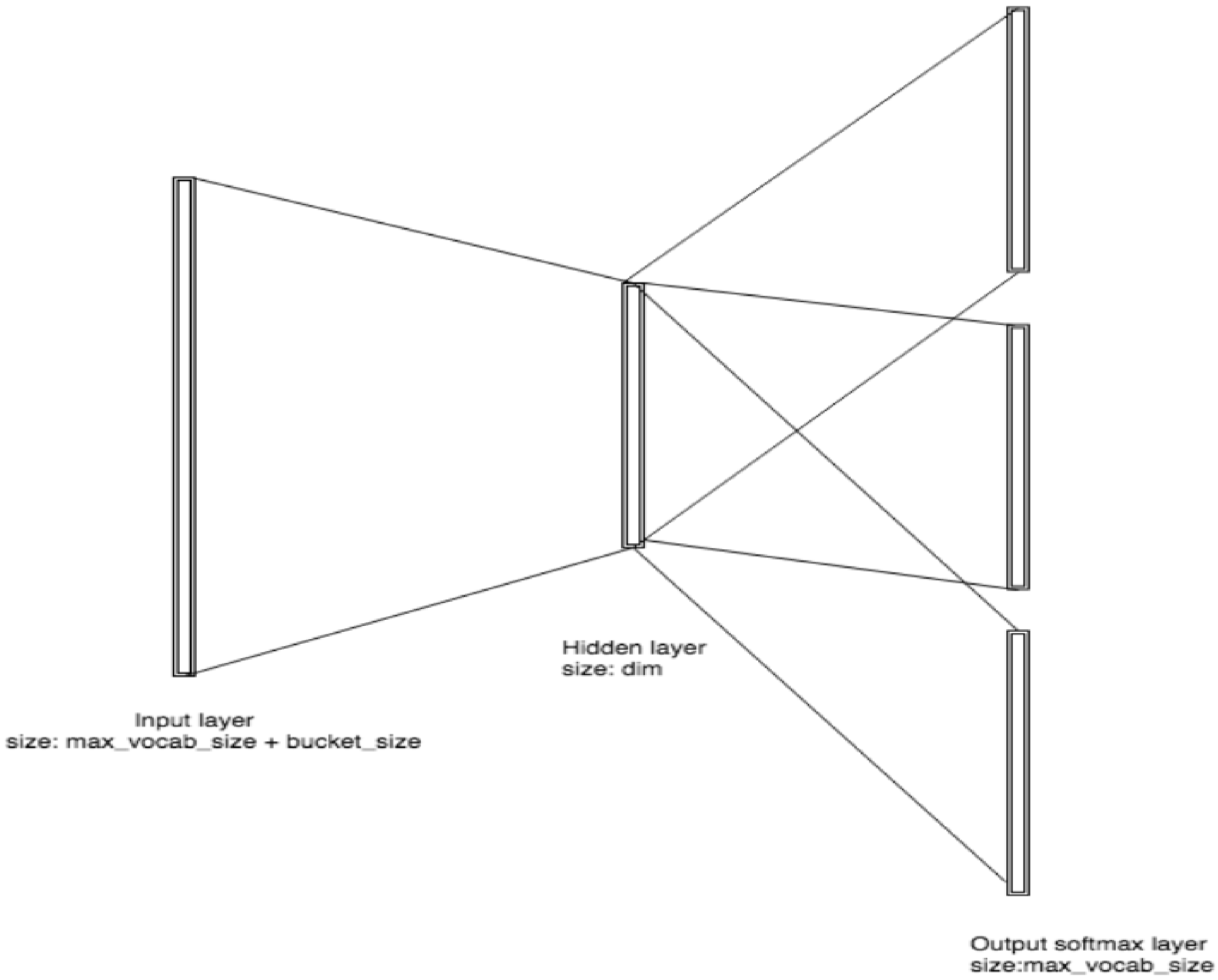
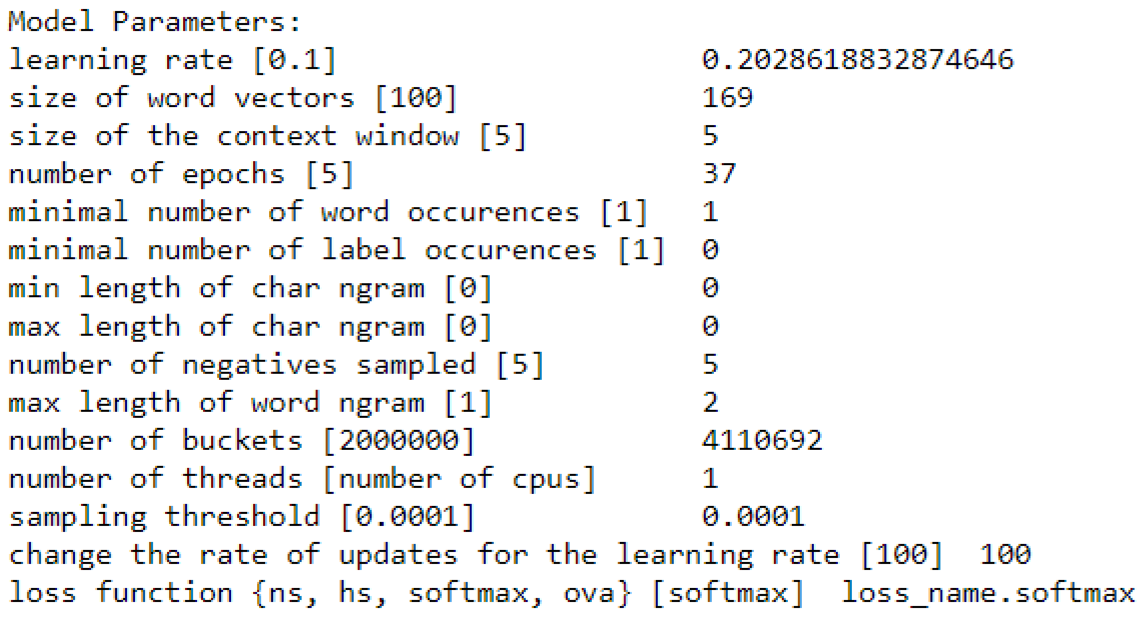
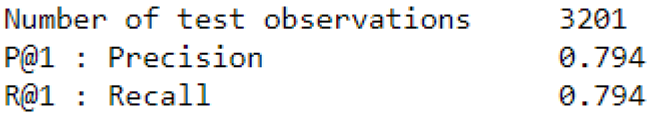


Figure 61: Fasttext model



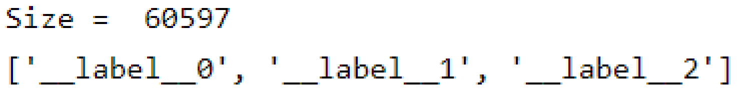
```
Model Parameters:
learning rate [0.1]                         0.2028618832874646
size of word vectors [100]                  169
size of the context window [5]              5
number of epochs [5]                        37
minimal number of word occurences [1]       1
minimal number of label occurences [1]      0
min length of char ngram [0]                0
max length of char ngram [0]                0
number of negatives sampled [5]             5
max length of word ngram [1]                2
number of buckets [2000000]                 4110692
number of threads [number of cpus]          1
sampling threshold [0.0001]                 0.0001
change the rate of updates for the learning rate [100]   100
loss function {ns, hs, softmax, ova} [softmax]  loss_name.softmax
```

Figure 62: Fasttext argument hyperparameters



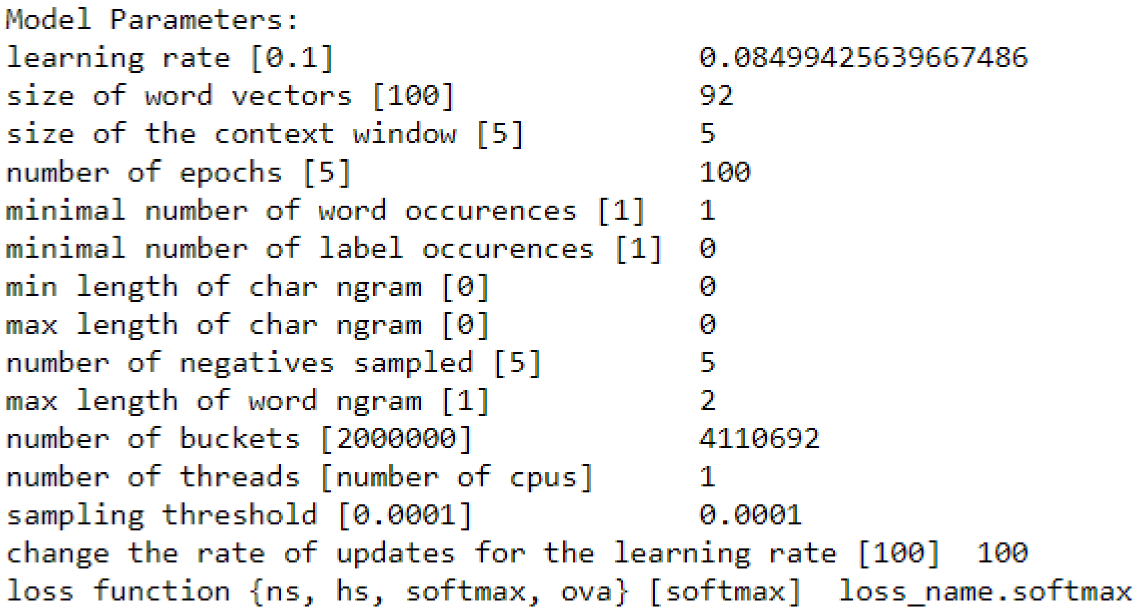
```
Number of test observations     3201
P@1 : Precision                 0.794
R@1 : Recall                    0.794
```

Figure 63: Fasttext argument metrics



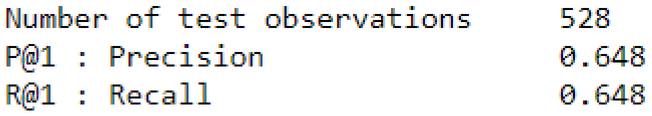
```
Size =  60597
['__label__0', '__label__1', '__label__2']
```

Figure 64: Fasttext argument size



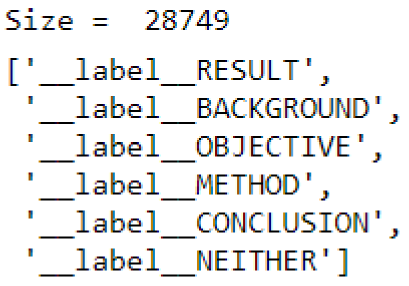
```
Model Parameters:
learning rate [0.1]                     0.08499425639667486
size of word vectors [100]              92
size of the context window [5]          5
number of epochs [5]                    100
minimal number of word occurences [1]   1
minimal number of label occurences [1]  0
min length of char ngram [0]            0
max length of char ngram [0]            0
number of negatives sampled [5]         5
max length of word ngram [1]            2
number of buckets [2000000]             4110692
number of threads [number of cpus]      1
sampling threshold [0.0001]             0.0001
change the rate of updates for the learning rate [100]  100
loss function {ns, hs, softmax, ova} [softmax]  loss_name.softmax
```

Figure 65: Fasttext structure hyperparameters



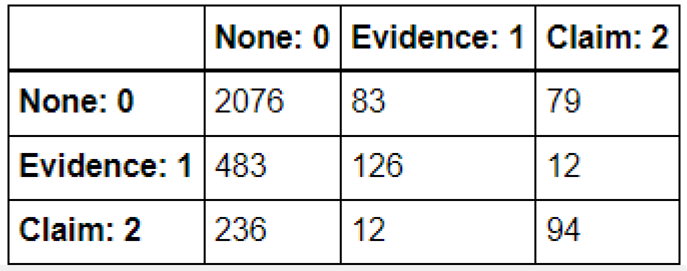
```
Number of test observations     528
P@1 : Precision                 0.648
R@1 : Recall                    0.648
```

Figure 66: Fasttext structure metrics



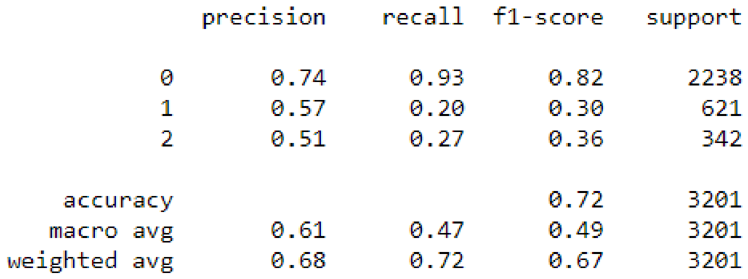
```
Size =  28749
['__label__RESULT',
 '__label__BACKGROUND',
 '__label__OBJECTIVE',
 '__label__METHOD',
 '__label__CONCLUSION',
 '__label__NEITHER']
```

Figure 67: Fasttext structure size



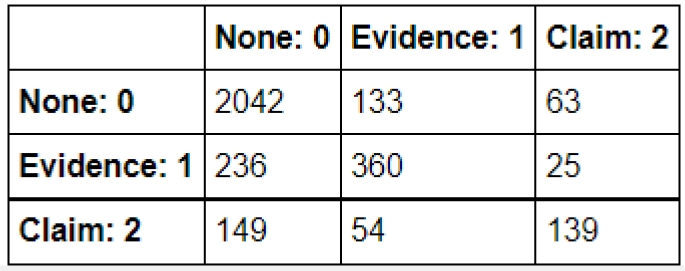
	None: 0	Evidence: 1	Claim: 2
None: 0	2076	83	79
Evidence: 1	483	126	12
Claim: 2	236	12	94

Figure 68: Confusion Matrix baseline argument



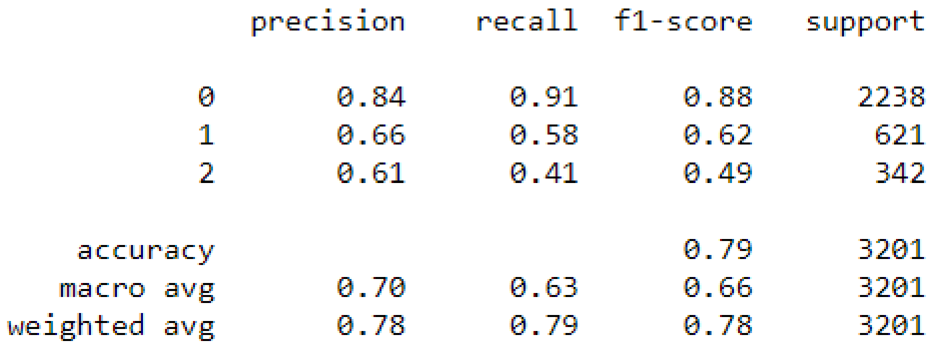
```
              precision    recall  f1-score   support

           0       0.74      0.93      0.82      2238
           1       0.57      0.20      0.30       621
           2       0.51      0.27      0.36       342

    accuracy                           0.72      3201
   macro avg       0.61      0.47      0.49      3201
weighted avg       0.68      0.72      0.67      3201
```

Figure 69: Classification report baseline argument



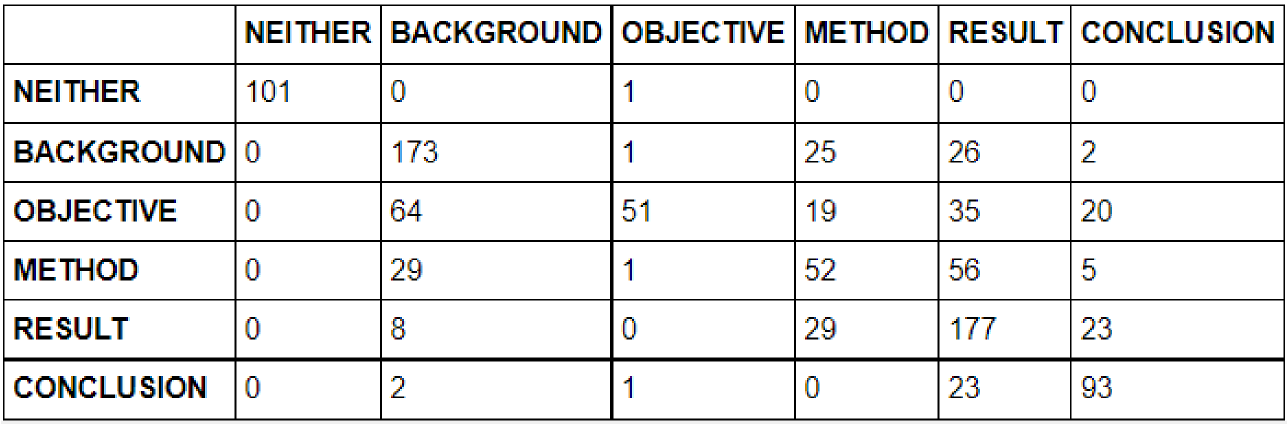
	None: 0	Evidence: 1	Claim: 2
None: 0	2042	133	63
Evidence: 1	236	360	25
Claim: 2	149	54	139

Figure 70: Confusion Matrix fasttext argument



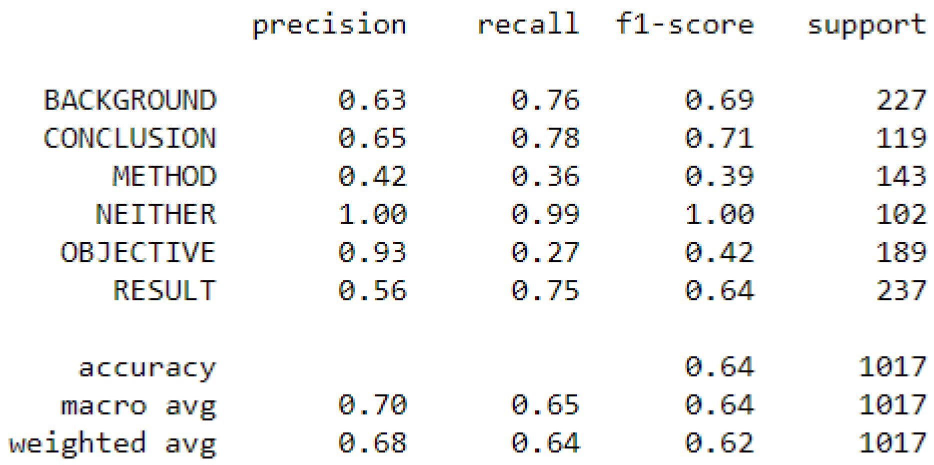
```
              precision    recall  f1-score   support

           0       0.84      0.91      0.88      2238
           1       0.66      0.58      0.62       621
           2       0.61      0.41      0.49       342

    accuracy                           0.79      3201
   macro avg       0.70      0.63      0.66      3201
weighted avg       0.78      0.79      0.78      3201
```

Figure 71: Classification report fasttext argument



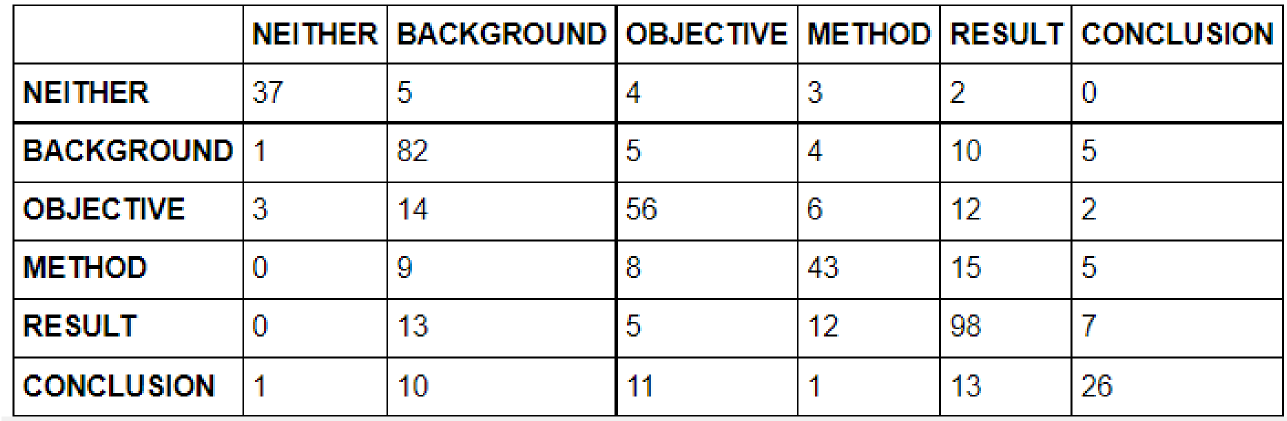
	NEITHER	BACKGROUND	OBJECTIVE	METHOD	RESULT	CONCLUSION
NEITHER	101	0	1	0	0	0
BACKGROUND	0	173	1	25	26	2
OBJECTIVE	0	64	51	19	35	20
METHOD	0	29	1	52	56	5
RESULT	0	8	0	29	177	23
CONCLUSION	0	2	1	0	23	93

Figure 72: Confusion Matrix for baseline structure model



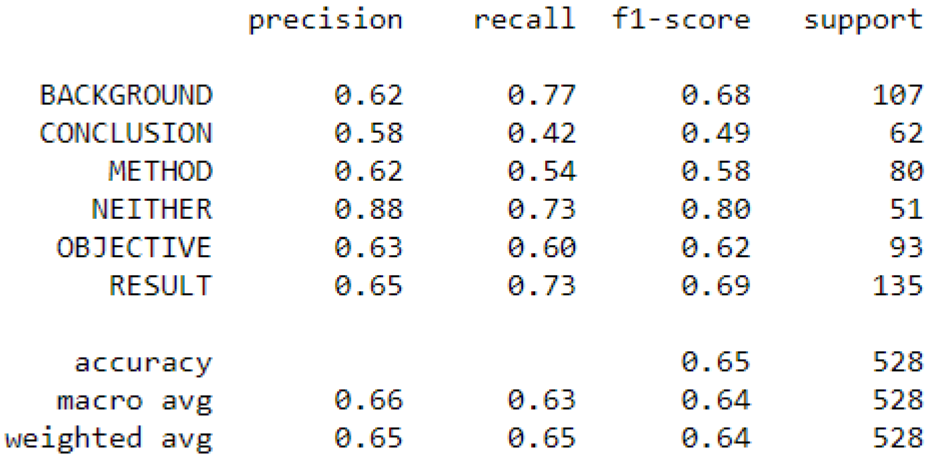
```
              precision    recall  f1-score   support

  BACKGROUND       0.63      0.76      0.69       227
  CONCLUSION       0.65      0.78      0.71       119
      METHOD       0.42      0.36      0.39       143
     NEITHER       1.00      0.99      1.00       102
   OBJECTIVE       0.93      0.27      0.42       189
      RESULT       0.56      0.75      0.64       237

    accuracy                           0.64      1017
   macro avg       0.70      0.65      0.64      1017
weighted avg       0.68      0.64      0.62      1017
```

Figure 73: Classification report for baseline structure model



	NEITHER	BACKGROUND	OBJECTIVE	METHOD	RESULT	CONCLUSION
NEITHER	37	5	4	3	2	0
BACKGROUND	1	82	5	4	10	5
OBJECTIVE	3	14	56	6	12	2
METHOD	0	9	8	43	15	5
RESULT	0	13	5	12	98	7
CONCLUSION	1	10	11	1	13	26

Figure 74: Confusion Matrix for fasttext structure model



```
              precision    recall  f1-score   support

  BACKGROUND       0.62      0.77      0.68       107
  CONCLUSION       0.58      0.42      0.49        62
      METHOD       0.62      0.54      0.58        80
     NEITHER       0.88      0.73      0.80        51
   OBJECTIVE       0.63      0.60      0.62        93
      RESULT       0.65      0.73      0.69       135

    accuracy                           0.65       528
   macro avg       0.66      0.63      0.64       528
weighted avg       0.65      0.65      0.64       528
```

Figure 75:Classification report for fasttext structure model

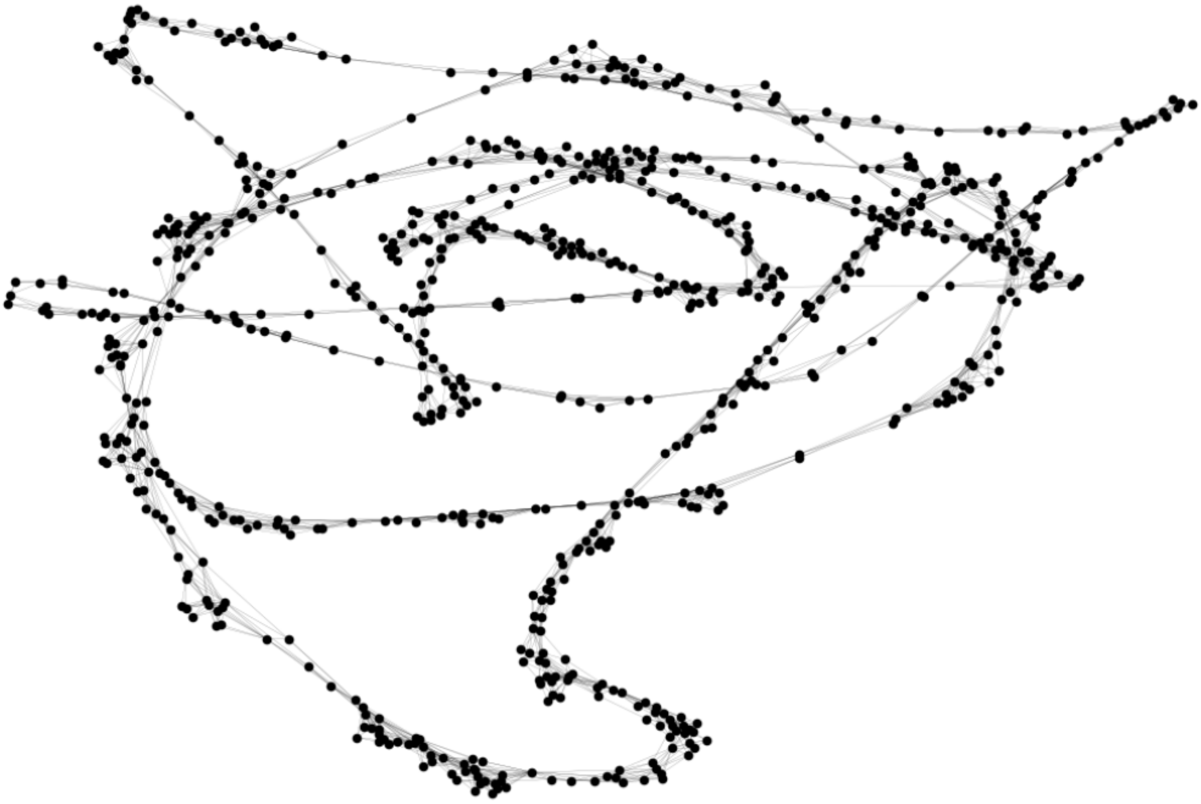


Figure 76:Graph

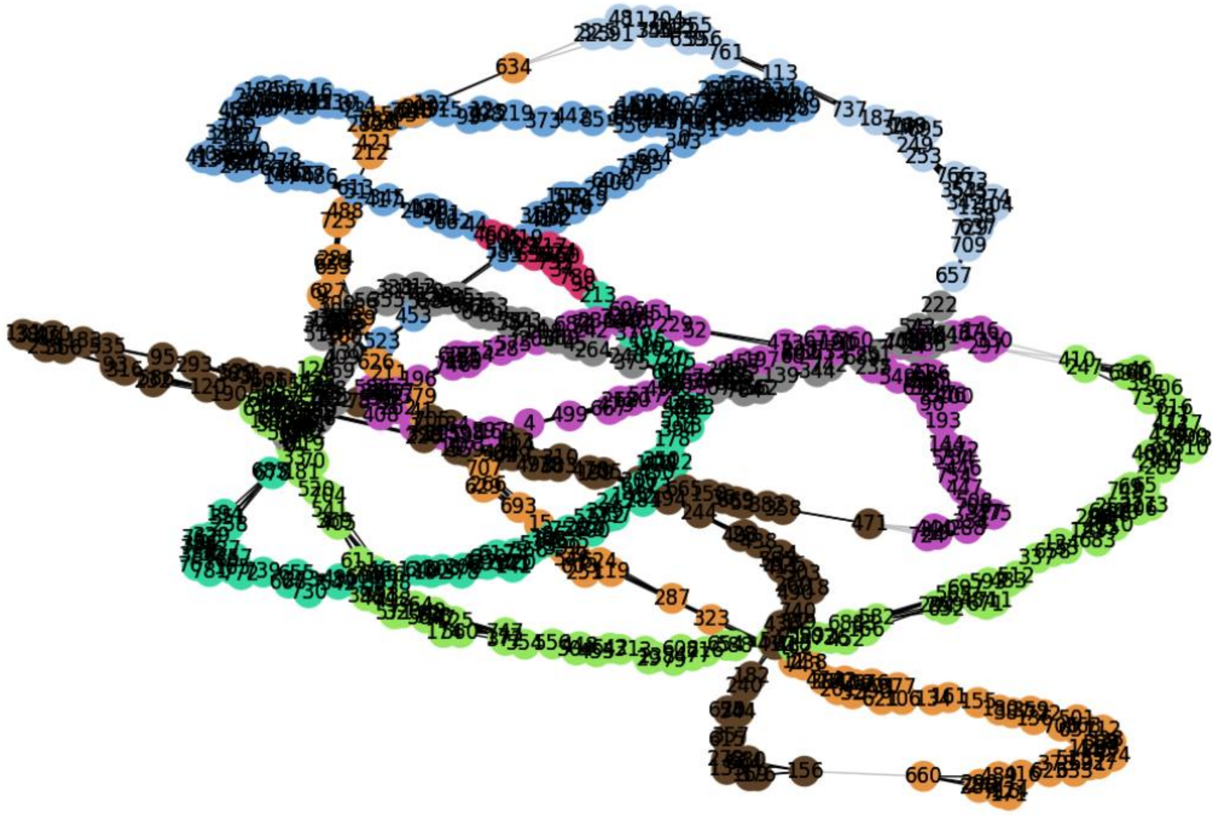


Figure 77: Clauset-Newman-Moore communities

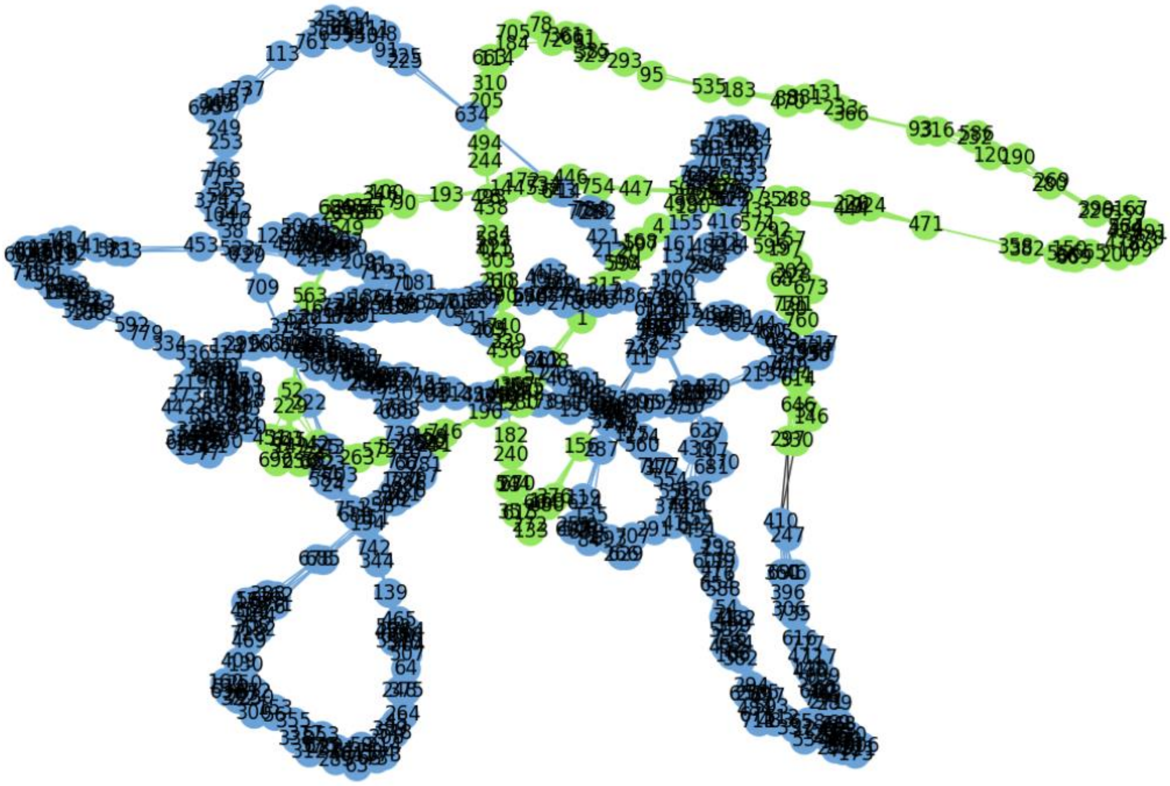


Figure 78:Girvan-Newman Community Detection

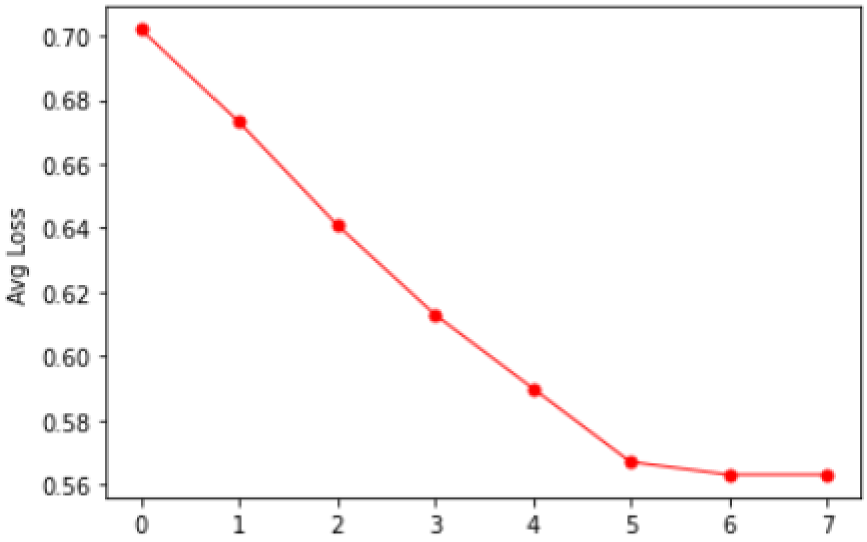


Figure 79: Avg loss plot fasttext argument

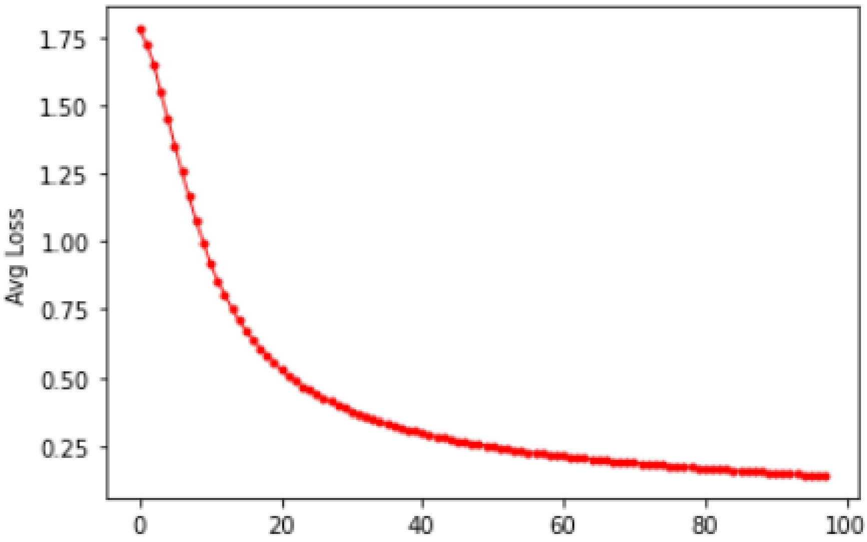
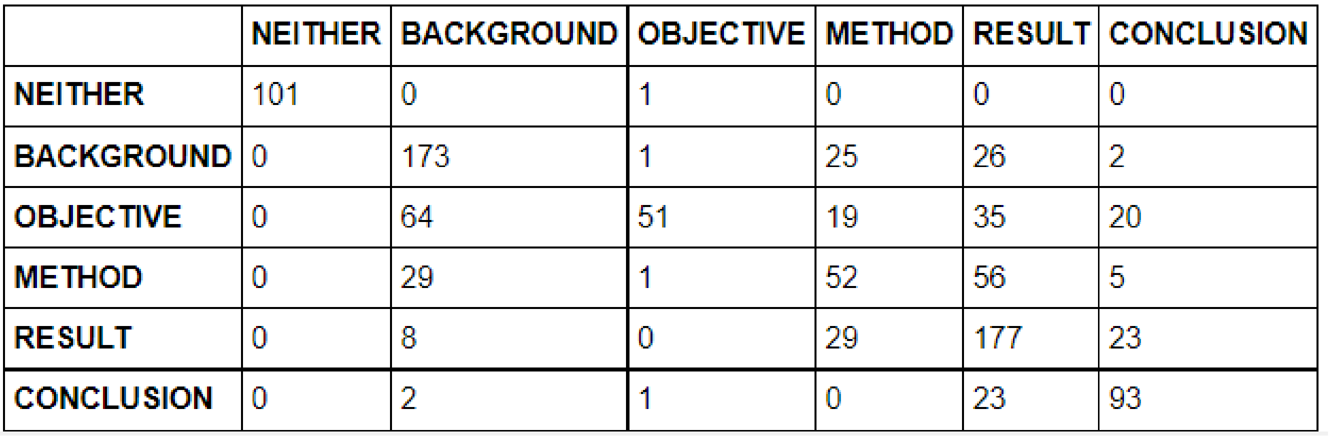


Figure 80: Avg loss plot fasttext structure

	Fasttext		
Model	TRAINING	VALIDATION	TESTING
ARGUMENT	0.99	0.78	0.79
STRUCTURE	0.99	0.64	0.65

Figure 81: Table of precision/recall for models



	NEITHER	BACKGROUND	OBJECTIVE	METHOD	RESULT	CONCLUSION
NEITHER	101	0	1	0	0	0
BACKGROUND	0	173	1	25	26	2
OBJECTIVE	0	64	51	19	35	20
METHOD	0	29	1	52	56	5
RESULT	0	8	0	29	177	23
CONCLUSION	0	2	1	0	23	93

Figure 82: Confusion matrix baseline structure