Final task - Same side classification

Lukas, Thang, Annie, Ruta

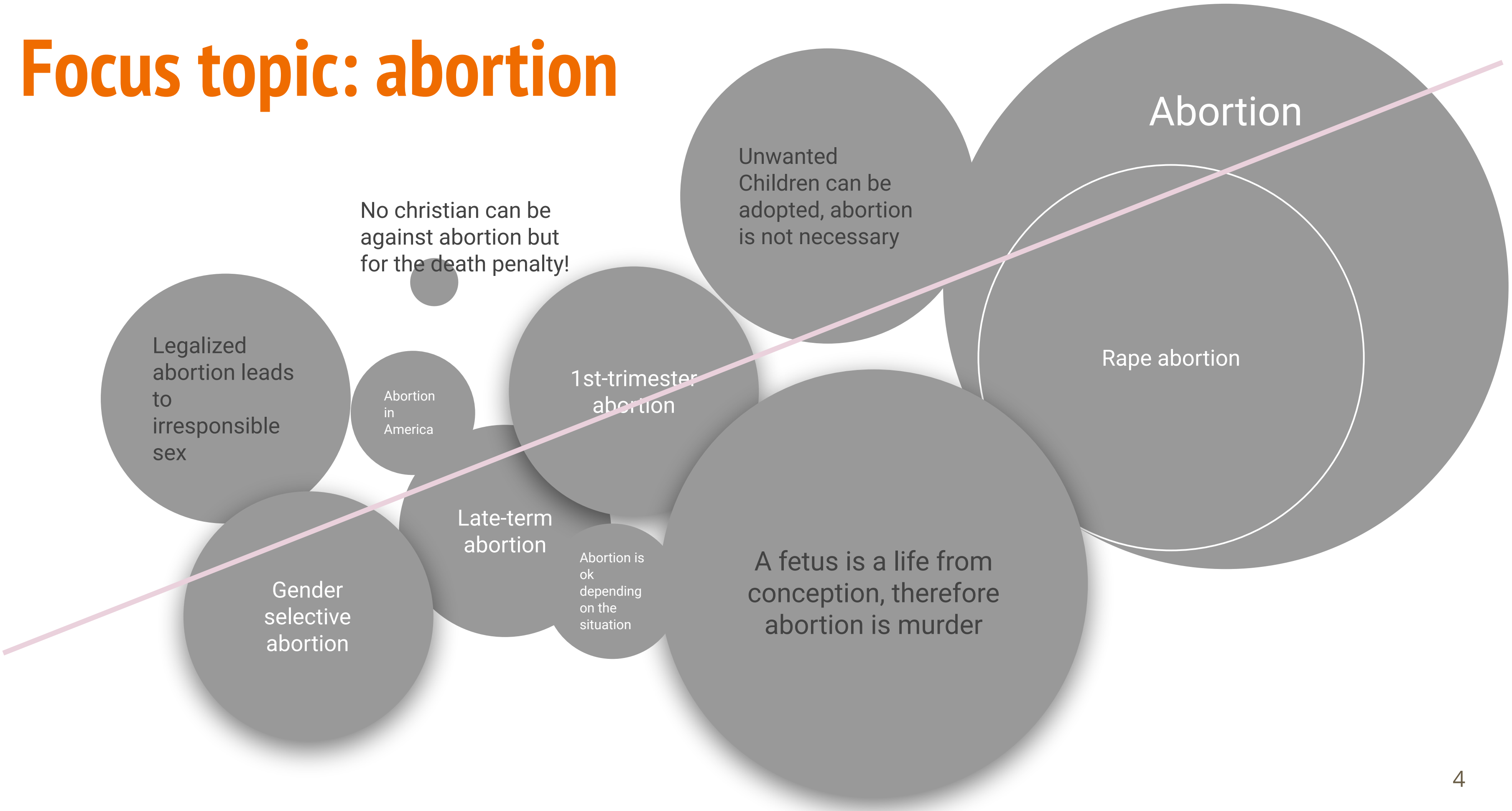
Same-side classification

- To identify the stance of a statement / argument towards a topic,
- => Combine pairs of statements / arguments and check whether they are in the same / different side of stance.
- This is pairs of text classification problem
  - More advanced than a text classification problem
- Our approach:
  - Stance classification
  - Mixed supervised learning
  - Explicit semantic analysis
  - Hierarchical attention network
- Different results on different datasets

Stance classification

- Original question of same-side classification
  - for & for, against & against => same side
  - for & against => different side
- args.me dataset, focus on one topic: abortion

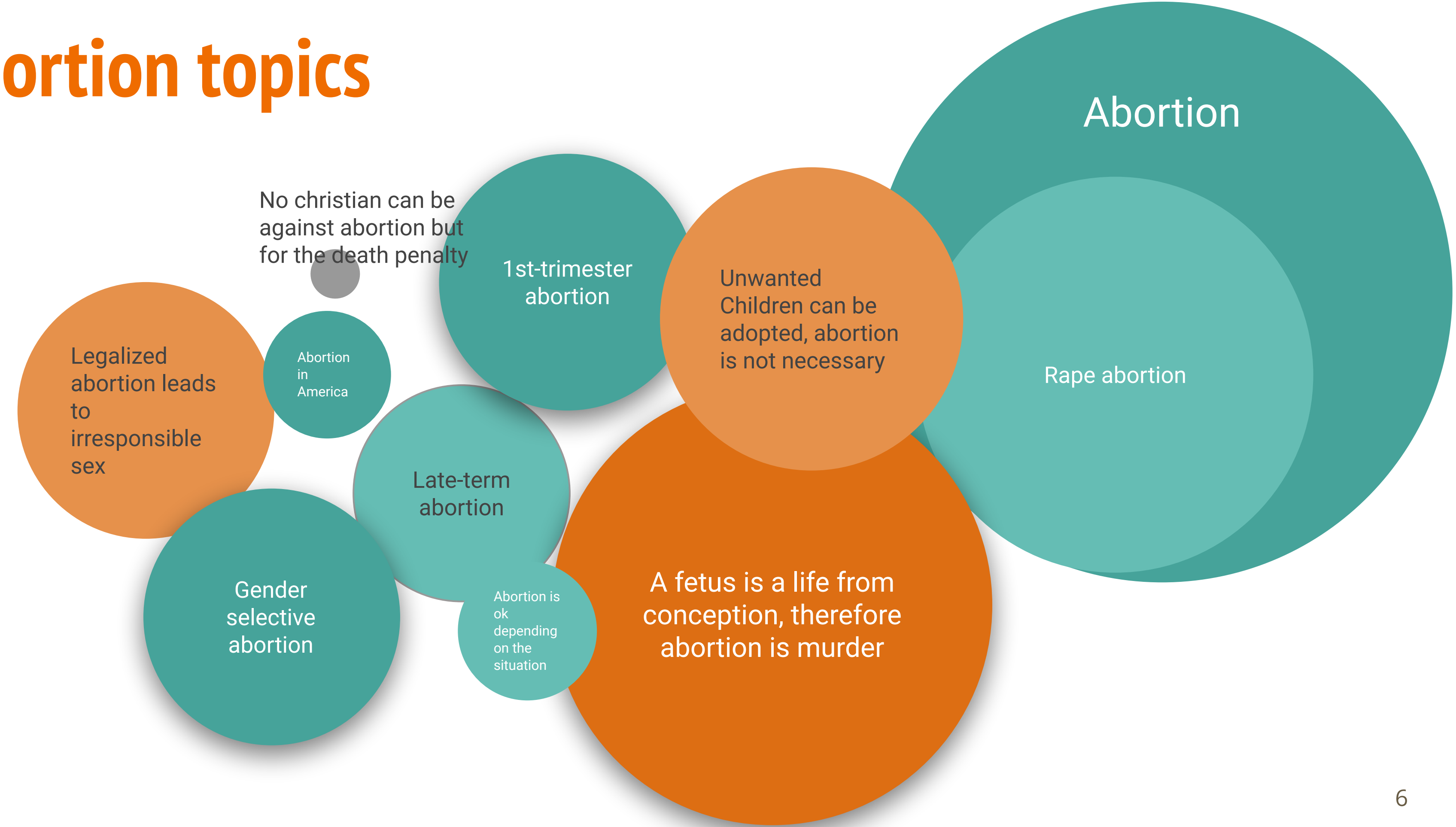
Focus topic: abortion



Abortion topic separator

- Consider abortion
  - Happens at all time (e.g first trimester, late term)
  - For all reasons (e.g rape, gender selection)
  - Label irrelevant topics
- Manually label stance of the **topic** (2 times)
  - First, label the topics
  - Second, check all the arguments if topic is correctly labelled
- 335 topics

Abortion topics



Observation for args.me dataset

- Debaters make arguments differently, in term of
  - Length
  - Rationality
  - Style
  - Source (law, bible, books, ...)
  - (sometimes) On wrong side of topic
  - (often) Make parallel argument (no conclusion)
  - Typos
- They often quote the opponents' arguments in " "
  - Can confuse stance classifier
  - Can help same-side classifier

Naive approach - similarity comparison

- Initial idea:
  - Extract claims from arguments - or Summarize arguments
  - Generate all phrases regarding abortion
  - Classify stance based on similarity comparison

Data Cleaning

- Lowercase text
- Remove hyperlink
- Decontract (I’ve -> I have...)
- Remove content in double quote, square bracket
- Remove special character except space
- Remove words containing number
- Remove cliches like vote pro, vote con
- Remove short sentences (< 4 words)

Summarization of arguments

- Compute cosine similarity between sentences in argument
- Build a ranking based on similarity matrix
- Take the one-fifth of all sentences from the ranking

Sorry, I meant to write "One" not "on". That's tablets for you. But anyways, the first trimester is the best time to abort the pregnancy as all the embryo is at this point is a ball of stem cells. It isn't it's own life yet and the embryo wouldn't be conscious until it developed a brain which happens later into the pregnancy. Aborting now is the perfect time to do so because of that. You're not ending a life at this point to suit yourself, all you're doing is preventing your pregnancy and Then again how can something be illegal if you don't get punished for being caught doing it? The whole title makes no sense in this case. It would be the same way if it was legal. Maybe you're on about having doctors give permissions for abortions? Is that what you mean? If so, that's one way it can be done. Otherwise, I can't think about anything else because there isn't really a law in this hypothetical scenario that actually prevents on demand abortions as this isn't investigated nor punished. There has to be a sanction for breaking a law whether that's a fine or a prison sentence. The whole question makes no sense!

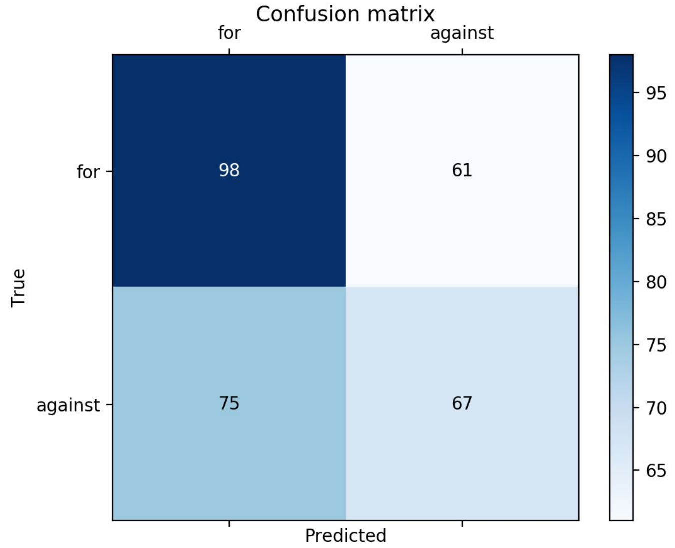
→

It is not it is own life yet and the embryo would not be conscious until it developed a brain which happens later into the pregnancy. There has to be a sanction for breaking a law whether that is a fine or a prison sentence. But anyways the first trimester is the best time to abort the pregnancy as all the embryo is at this point is a ball of stem cells.

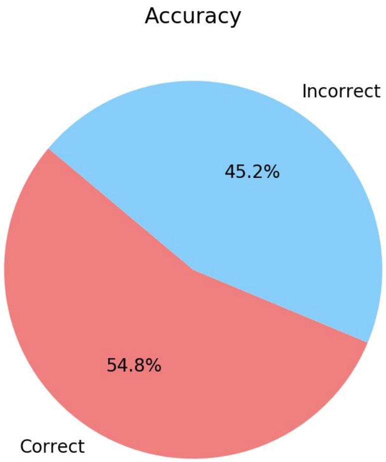
Using naive approach

- Split into 80-20 training and test dataset
- Training set:
    - Split arguments into sentences
    - Label all sentences: for and against
- Test set:
    - Split arguments into sentences
    - For each sentence, using cosine similarity to find the closest sentence in training set
    - Label that sentence accordingly
    - Label the argument based on dominant label

Results



Confusion matrix
for
against
True
for
against
98
61
75
67
Predicted
95
90
85
80
75
70
65



Accuracy
Incorrect
45.2%
54.8%
Correct

Why naive approach does not work?

- Noisy data
  - Not all sentences are related to abortion `thank you for posting i look forward to it`
  - Sentences are too long `23 tokens / sentence`
  - Similarity calculated based on lexical not semantic
  - Different writing style, vocabulary, …
- High cosine similarity does not mean same side of arguments

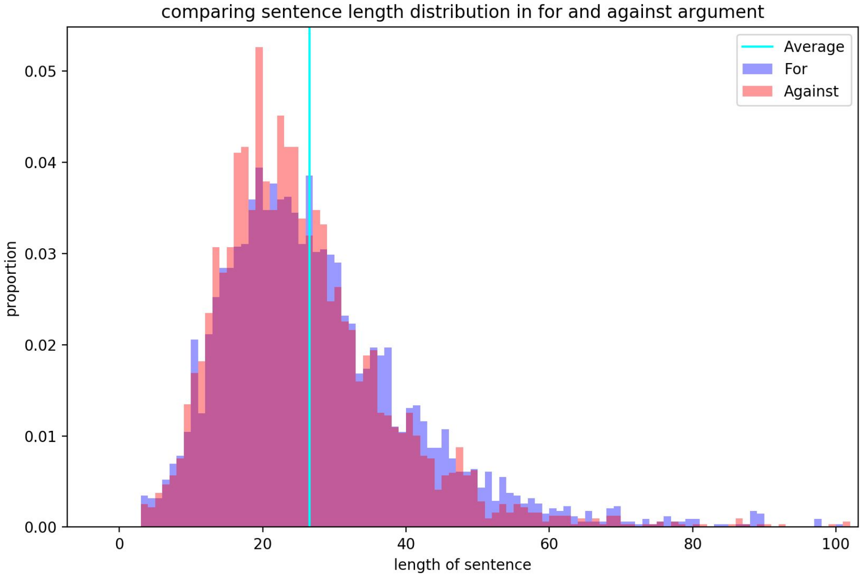
  Abortion funds | Abortion out!

  Abortion is a good thing | Abortion is murder
- Suggestion
  - Use POS to divide the sentence into smaller clause
  - Use negativity cues to check the side of the clause
- Other methods?

More approaches

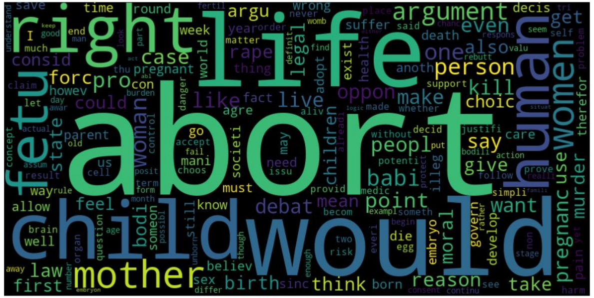
- Clauses / Phrases analysis
- Bag of Words
- Tfidf
- RNN => LSTM + pretrained embedding
- => Which is more suitable?
  - Implement Exploratory Data Analysis

Sentence length

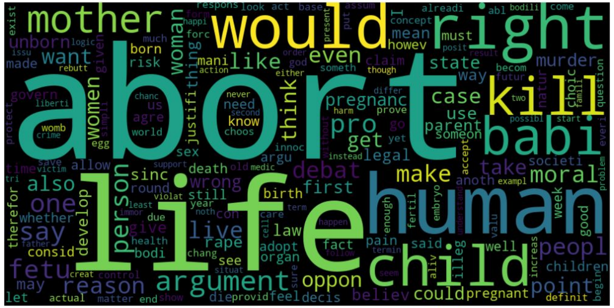


1-gram token distribution

for unigrams



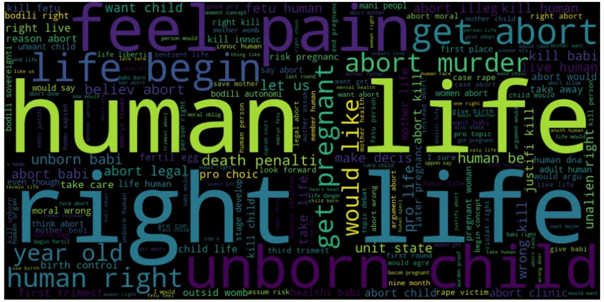
against unigrams



2-gram token distribution

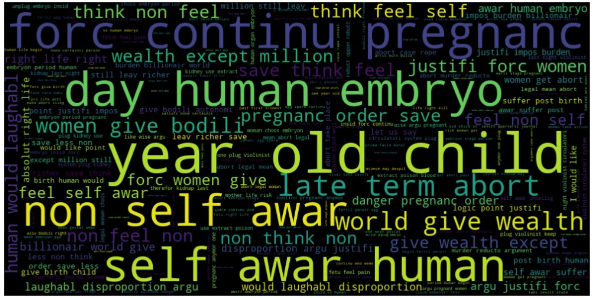
for bigrams

against bigrams

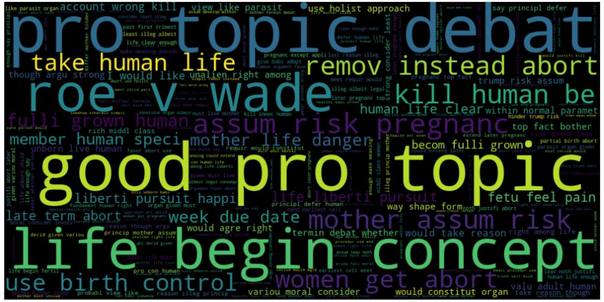


3-gram token distribution

for trigrams

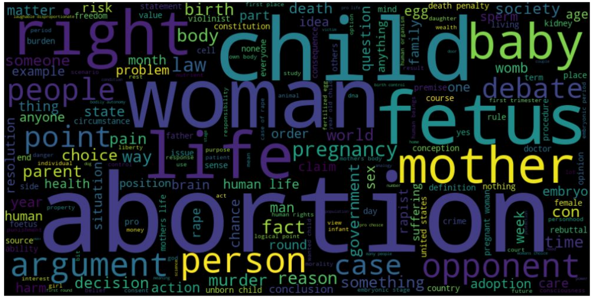


against trigrams

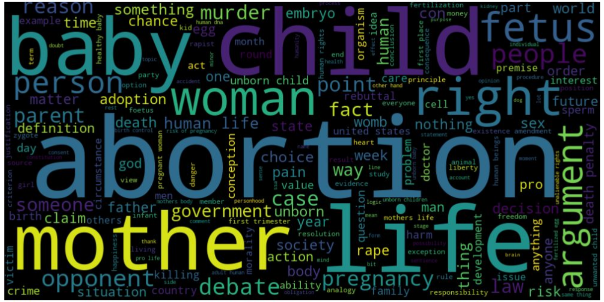


Phrases distribution

for abortion phrases

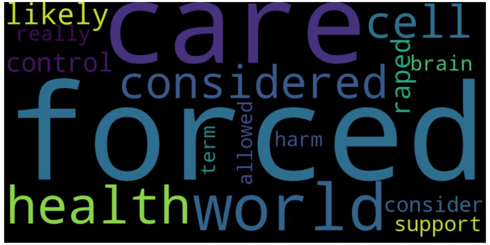


against abortion phrases



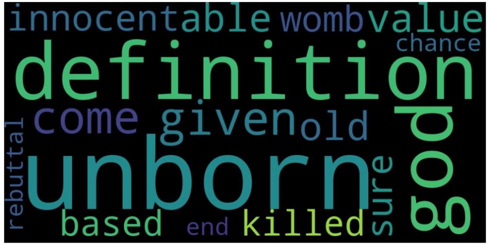
Tfidf distribution

for abortion tokens



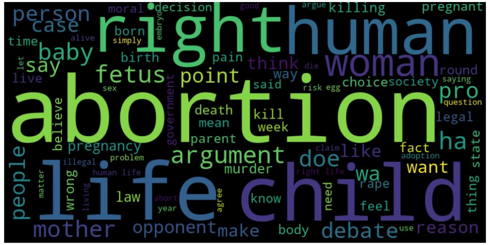
Distinct

against abortion tokens



Distinct

common abortion tokens

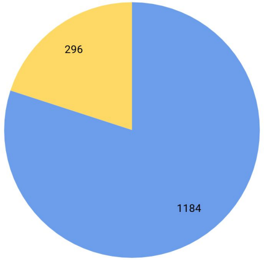


Common

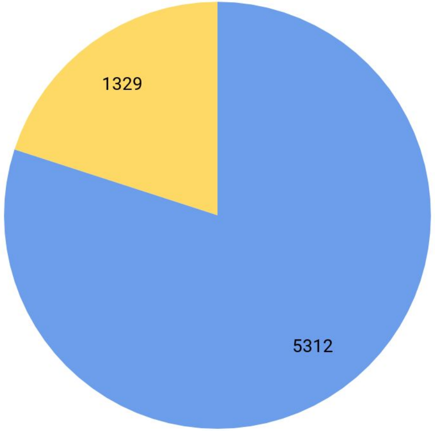
Next step?

- Deduction from analysis
    - BoW may not work
    - Sequences may work
- => Deep Learning: Glove pre-trained embedding 300 dimension (keep semantic meaning) + Train with RNN / LSTM and test
- Test set is in same topic (abortion) with training set
    - In-domain (argsme)
    - Cross-domain (argsme vs parliament)
    - Filter out parliament statements which talk about abortion (62) and manually label them (for, against, no-stance)

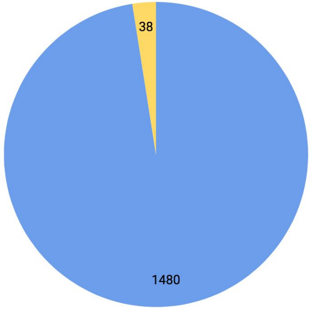
Train-test ratio



In-domain (argument-based)



In-domain (sentence-based)



Cross-domain

Why this model does not work?

- The model is too simple
- Dataset is too small
    - ~1500 data is not enough for a deep learning model
- Secondary labelling is not effective
    - Each argument focuses on a deeper topic
    - Argument from same topic maybe related to each other
    - But arguments in different topic are not
- Non-consistent dataset:
    - Debaters have different argumentation and writing style
    - Noisy and informal text

Supervised learning for stance classification

Goal :

Obtaining stance of the speeches of Canadian parliament and listing them as *for* and *against* the topic in args.me

Hypothesis :

For any given term of Canadian parliament, for any given topic and for any political party, speeches made by the members of the given party have the same stance.

General Idea

- Sentence Similarity :
  - Semantic similarity : The distance between an argument pair based on the meaning or semantic content
  - Word order similarity : Similarity between order of words in a sentence

- Sentiment Analysis :
  - Check the similarity between polarity of arguments

- Semantic Similarity :
    - Pre-trained method (Glove - 300d)
    - Cosine Similarity
- Words are converted into numerical vectors using Glove. It is better than tf-idf because instead of assigning numbers, it assigns numeric vector to each word.
- This word vectors will be close in space if they have the same meaning.
- Computing the similarity between them

Advantage : Works well with arguments of different lengths as it measures the angle in the space and not the magnitude

- Word order similarity :
    - Using n-grams as features
    - Results are tested using 1-gram,2-gram and 3-gram features
    - 1-gram doesn’t capture the order where as 3-gram can be too specific. Whereas bi-grams helps to achieve better results.

- ***Advantage*** : Keeps track of word combinations or negates
- ***Disadvantage*** : More training data is required

- Sentiment Analysis :
    - Checks the polarity of the speech user VADER sentiment analysis
    - Checks if two arguments have the same polarity

Advantage : Works well with n-gram features

Data processing

- Convert to lower text
- Remove all the punctuations and hyper texts
- Tokenization
- Removal of stop words
- Lemmatization
- Vectorization of data using tf-idf

Stance classification approach

Goal : To find stance of the **single speech** with respect to topic

Dataset : Annotated parliament data by Thang

Total : 555 favor : 478 against: 77

Approach : Applying suggested approach on speech text vs. topic using random forest with grid search classifier

Accuracy : 66.84 %

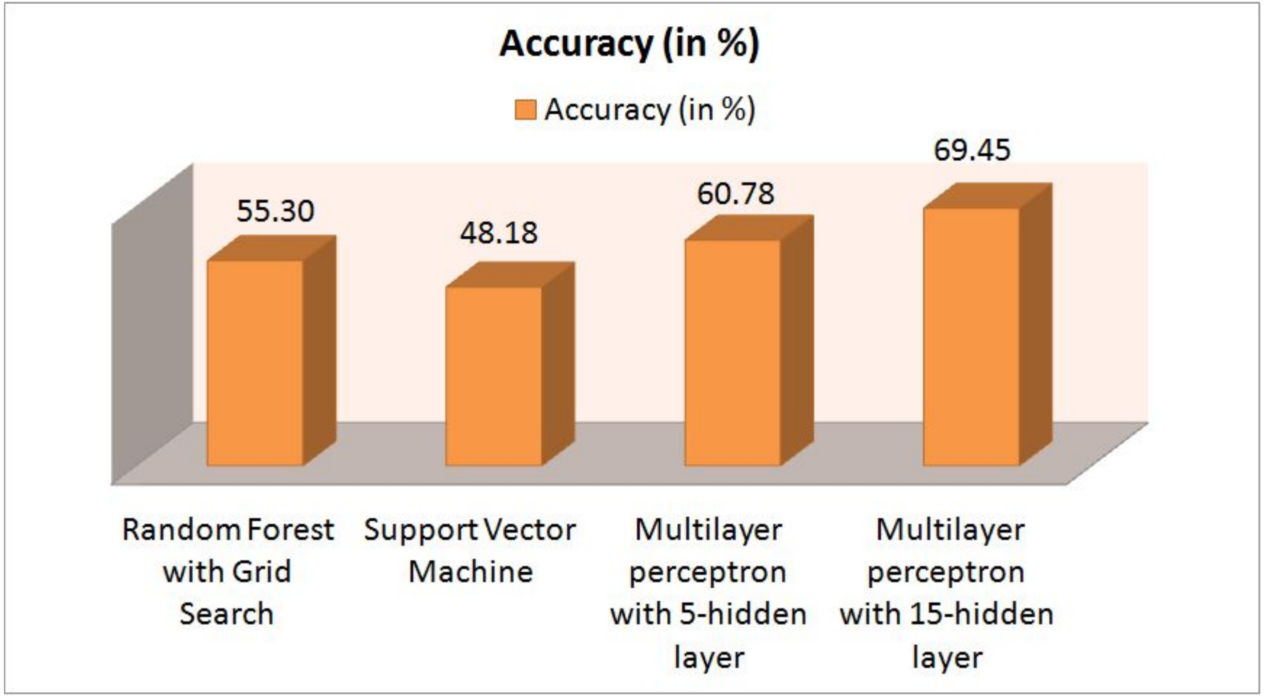
Disadvantage : Requires large number of labeled data with same topic, same political term and same party

Approach : Same side stance classification

- Data set : Webis same stance classification data
- Split into 66-33 % train and test data set

Total	Same Stance	Different stance
63886	34104	29782

Results comparison



Hypothesis testing dataset

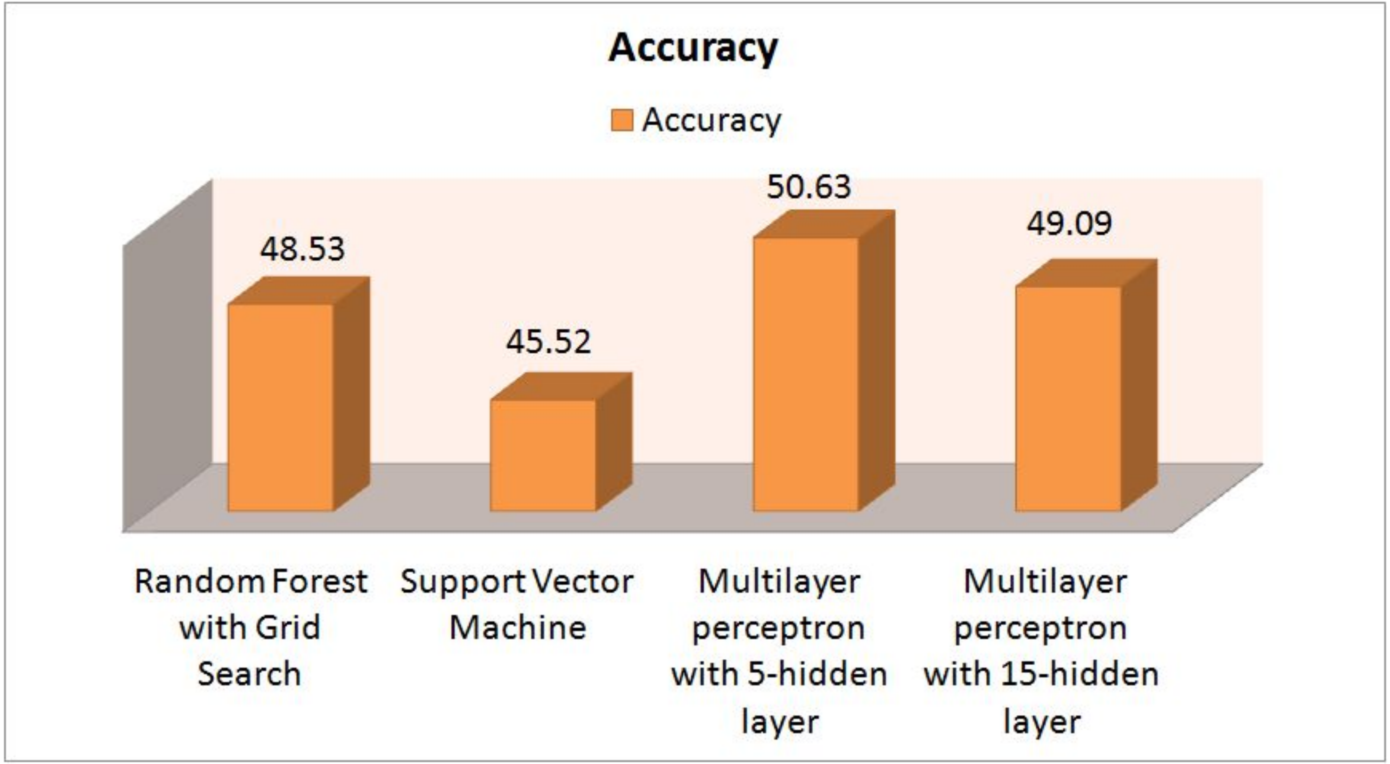
Dataset : Canadian parliament data argument pairs. Arguments pairs were made by taking political term, political party and the topic into consideration.

Total argument pairs	Positive pairs	Negative pairs
84752	53831	30921

Positive pairs : Same political terms

Negative pairs : Different political terms

Hypothesis testing



Reflection on different classifier performance

Support vector machine (SVM) :

- Linear SVM is not suitable when data is not linearly separable
- Choosing right kernel can be tedious because it can be computationally complex with increasing dimensionality of the data

Random forest with grid search :

- Does implicit feature selection
- Not too sensitive to hyper parameters settings
- Grid search helps to find the optimal parameters for most accurate predictions
- *Disadvantage :* Computationally expensive and time consuming

Multilayer perceptron :

- Works well with sequential data
- Very flexible and can be applied to other types of data as well
- ***Disadvantage :***
    - Sensitive to hyper parameters
    - May overfit with more number of hidden layers

Conclusion :

Assumed hypothesis is not significantly true!!

- For the same topic, there can be many different subtopics are being discussed and stance of the parties can vary.
  - For example, "*Olympic Games*",
  - There could be various issues related to Olympic games such as budget allocation, effect on environment, safety of the public etc.
  - So parties could have same stance on one sub topic (such as budget) but they could have different stance on another sub topic (safety)
- Using opponent's' argument in speech to counter support own opinion
- Different kinds of speaking styles such as sarcasm or humor

Same-Side Classification with ESA

Idea:

Using the ESA (explicit semantic analysis) representation of arguments to predict if they have the same stance

Same-Side Classification with ESA

ESA:

- We have a collection of n documents, where each document makes up a concept, that is represented by the terms in this document
- We then compute the tf-idf-weight for each term in each concept

	$concept_1$	...	$concept_n$
$term_1$	w_{11}	...	w_{n1}
...	...	...	...
$term_m$	w_{m1}	...	w_{mn}

Same-Side Classification with ESA

An argument is represented as its term-frequency vector, only terms that appear also in the ESA-matrix are considered.

We then compute the ESA-representation of the argument, by computing the scalar product between its vector-representation and the column (concept) vectors of the ESA-matrix.

The ESA-representation of an argument is then an n-dimensional vector where each entry represents how strong an argument belongs to the respective concept.

Same-Side Classification with ESA

To tackle the same-side-classification task, ESA with two concepts was used, one concept represents the stance “pro” and the other “con”.

To classify, if two arguments have the same stance we then can :

- Compute the cosine-similarity of the ESA-Representations of the two arguments and consider arguments to have the same stance when their similarity is above a certain threshold
- Consider the stance of an argument to be the stance that has the largest value in their ESA-representation and predict if arguments have the same stance based on that

Same-Side Classification with ESA

Experiment 1:

- Evaluation on the **in-domain same-side-classification** training-set
- For the construction of the **ESA-matrix** arguments from the **args.me corpus** (without arguments from debate.org) were used, that consider the topic **“abortion”** or **“gay marriage”**
- The arguments with the stance **“pro” make up one concept** in the ESA-matrix and the **“con” arguments the other**

Same-Side Classification with ESA

- 34111 (**53.3 %**) pairs that have the same stance
- 29792 (**46.4 %**) pairs that have a different stance

Classification by Maximum Value

Accuracy: **56.8 %**

F1-Score: **0.69**

	Same-Side	Not Same-Side
Same-Side	**30698**	**24178**
Not Same-Side	**3413**	**5614**

Classification by Cosine-Similarity

Accuracy: **59.2 %**

F1-Score: **0.64**

	Same-Side	Not Same-Side
Same-Side	**23596**	**15526**
Not Same-Side	**10515**	**14266**

Same-Side Classification with ESA

Classifier was biased towards predicting arguments as “pro”, so the weights for the concept “con” were increased by 0.01 in the ESA-matrix

Classification by Maximum Value

Accuracy: **59.4 %**

F1-Score: **0.65**

	Same-Side	Not Same-Side
Same-Side	**24546**	**16361**
Not Same-Side	**9565**	**13431**

Classification by Cosine-Similarity

Accuracy: **59.6 %**

F1-Score: **0.68**

	Same-Side	Not Same-Side
Same-Side	**27383**	**19060**
Not Same-Side	**6728**	**10732**

Same-Side Classification with ESA

Experiment 2:

- Evaluation on the argument pairs of the **in-domain same-side-classification** training-set with the **topic “gay marriage”**
- For the construction of the **ESA-matrix** arguments from the **args.me corpus** (without arguments from debate.org) were used, that consider the topic **“abortion”**
- The arguments with the stance **“pro” make up one concept** in the ESA-matrix and the **“con” arguments the other**

Same-Side Classification with ESA

- 13277 (**57.6 %**) pairs that have the same stance
- 9786 (**42.3 %**) pairs that have a different stance

Classification by Maximum Value
Accuracy: **56.1 %**
F1-Score: **0.68**

	Same-Side	Not Same-Side
Same-Side	**10993**	**7831**
Not Same-Side	**2284**	**1955**

Classification by Cosine-Similarity
Accuracy: **53.1 %**
F1-Score: **0.59**

	Same-Side	Not Same-Side
Same-Side	**7905**	**5444**
Not Same-Side	**5372**	**4342**

Same-Side Classification with ESA

Classifier was biased towards predicting arguments as “pro”, so the weights for the concept “con” were increased by 0.01 in the ESA-matrix

Classification by Maximum Value

Accuracy: **52.0 %**

F1-Score: **0.59**

	Same-Side	Not Same-Side
Same-Side	**7976**	**5754**
Not Same-Side	**5301**	**4032**

Classification by Cosine-Similarity

Accuracy: **54.8 %**

F1-Score: **0.66**

	Same-Side	Not Same-Side
Same-Side	**10330**	**7473**
Not Same-Side	**2947**	**2313**

Same-Side Classification with ESA

Experiment 3:

- Evaluation on pairs of **speeches**, where speeches that took place during the **same period** of time and are from speakers of the **same party** are considered to have the **same stance**
- For the construction of the **ESA-matrix** arguments from the **args.me corpus** (without arguments from debate.org) that consider **any topic**
- The arguments with the stance **“pro” make up one concept** in the ESA-matrix and the **“con” arguments the other**

Same-Side Classification with ESA

- 53831 (**63.5 %**) pairs that have the same stance
- 30921 (**36.5 %**) pairs that have a different stance

Classification by Maximum Value

Accuracy: **53.6 %**
F1-Score: **0.64**

	Same-Side	Not Same-Side
Same-Side	**35697**	**21146**
Not Same-Side	**18134**	**9775**

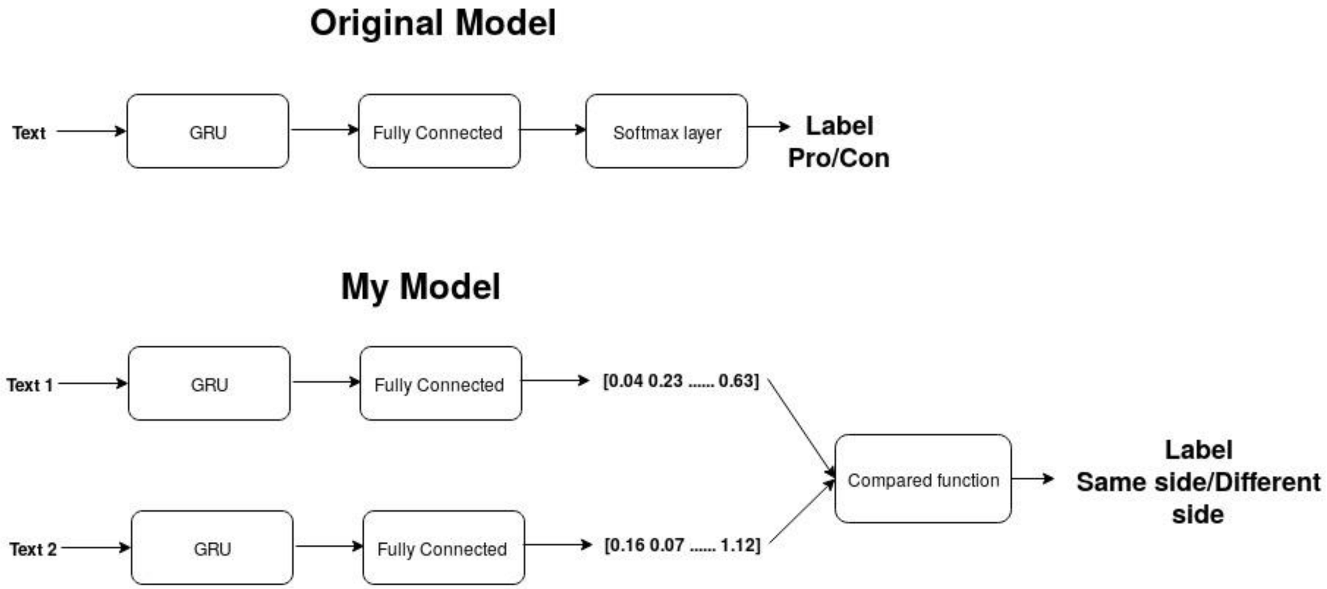
Classification by Cosine-Similarity

Accuracy: **44.8 %**
F1-Score: **0.39**

	Same-Side	Not Same-Side
Same-Side	**15183**	**8152**
Not Same-Side	**38648**	**22769**

Hierarchical attention network

The model is derived from Hierarchical Attention Network (HAN)



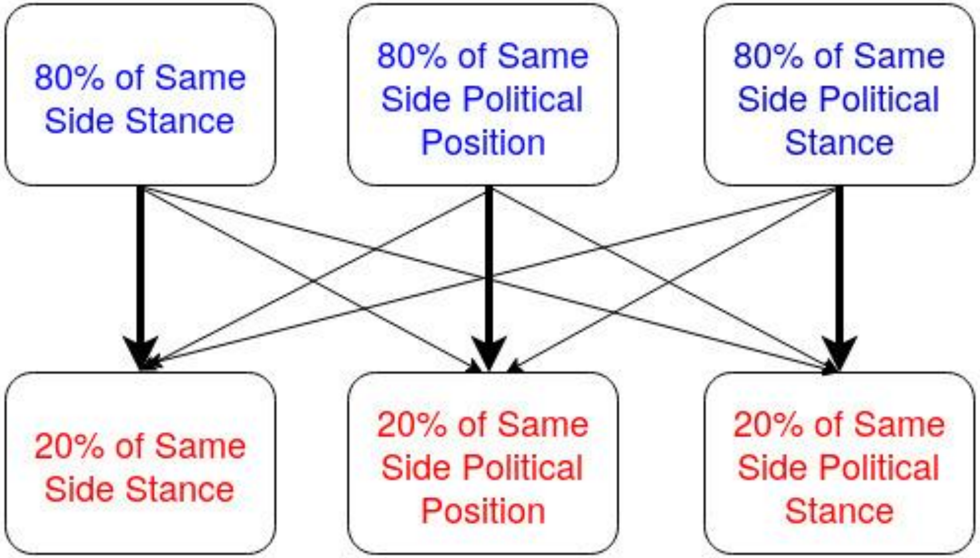
Dataset & Experiment

Dataset

1. **Same Side Stance**
   - Topic: Abortion, Gay Marriage
2. **Same Side Political Position**
   - Topic: 10 most frequent topics
3. **Same Side Political Stance**
   - Topic: Health, Taxation,

     Budget, Economy

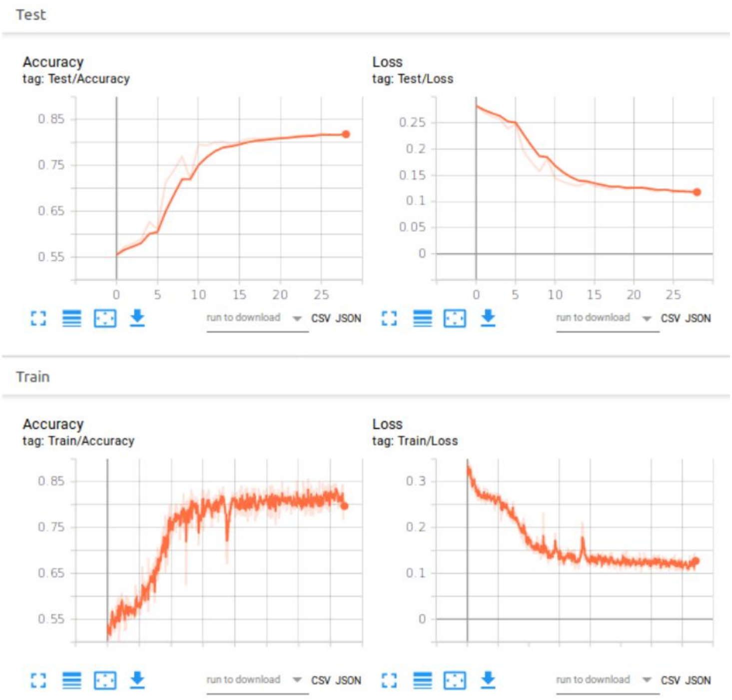
Experiment



Result (1)

Within domain (training and testing in **Same Side Stance** dataset)

Training + Validation



Testing

Accuracy
79.83%

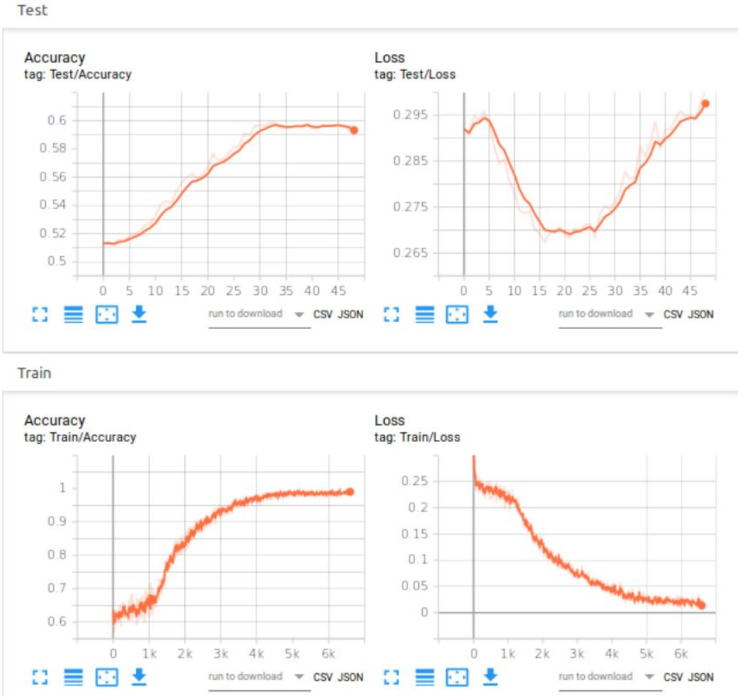
Detail statistics

	Precision	Recall	F1 score
Same side	85%	80%	82%
Different side	78%	83%	81%

Result (2)

Cross domain (training in **Same Side political position**, testing in **Same Side political stance**)

Training + Validation



Testing

Accuracy
57,12%

Detail statistics

	Precision	Recall	F1 score
Same side	56%	71%	63%
Different side	56%	40%	47%

Result (3)

Confusion matrix

Training and testing in **Same Side Stance** dataset

	Different side	Same side
Different side	4965 (38.85%)	994 (7.78%)
Same side	1382 (10.81%)	5440 (42.56%)

Training in **Same Side political position**, testing in **Same Side political stance**

	Different side	Same side
Different side	1717 (19.13%)	2616 (29.15%)
Same side	1334 (14.86%)	3308 (36.86%)

Explanation & Contribution

Explanation

Why the accuracy is not good in Cross-Domain experiment ?

1. Difference in datasets’ area
2. Difference in datasets’ speaker’s occupation
3. The gap between party side and opinion

Contribution

1. Same-side stance classification
2. Stance classification

Model's Output ?

What is the final output of training process ?

- Accuracy, Loss, Recall, Precision, F1 score ….. ?
- **Saved model**

Nobody want to repeat the process of preprocessing, training, testing again

- Loading the saved model to evaluate with unseen data

Future improvements

1. Using different pre-trained model (FastText....)
2. Using different loss function (L1, ...)
3. Using different optimization function (Adam, Rmsprop,....)
4. Evaluating this model with other dataset
5. Combining different neural network architecture (CNN, RNN, LSTM, ...)

General conclusion

- Same side classification is a hard task
- In domain testing results varies in different datasets
- argsme dataset has really different characteristics than parliamentary debate dataset
    - argsme text is from online users, with more informal text style
    - Parliamentary debate is from politicians, who always speak in formal with some standard structure
    - Cross domain testing results in a poor accuracy