

# Semantic Textual Similarity

Mauro Vázquez Chas, Dániel Mácsai

December 12, 2024

# Outline

- 1 Preprocessing
- 2 Features computed
- 3 Models and additional ideas
- 4 Course of Actions
- 5 Results
- 6 Takeaways

# Preprocessing

## Preprocessing

- Lowercasing and removing punctuation
- Tokenization with and without stopwords
- Lemmatization with and without stopwords
- Synsets computed by Lesk's algorithm

# Basic features

## Basic features

- Character ratio, token ratio
- Levenshtein ratio (edit distance)
- Jaccard similarity
  - on tokens with and without stopwords
  - on lemmas with and without stopwords
  - on synsets from Lesk's algorithm

# Utilizing synset similarity

## Best possible pair

For each word in each sentence, we get the best similarity value considering all of its possible synsets and all the synsets from the words in the other sentence, all of this without modifying its post tags. After this, take the average, considering only the tokens with a valid wordnet pos tag. Methods we used for the synset similarities:

- Path similarity
- Wu-Palmer similarity
- Leacock-Chodorow similarity

# Utilizing synset similarity

## Using Lesk's algorithm

We use Lesk's algorithm to match synsets to tokens in the two sentences. After, for each synset in a sentence, we compute the highest similarity value using the synsets from the other sentence, and we take the average of these similarities. Methods we used for the synset similarities:

- Path similarity
- Wu-Palmer similarity
- Leacock-Chodorow similarity

# Other features

## N-grams

- On tokens, lemmas and characters
- Using Jaccard similarity for sets and cosine similarities on histograms

## SentiWordNet

For both the positive and the negative scores we compute the following getting two features in total: For each sentence, we sum the scores of each Lesk synset. Afterwards, we subtract the values of the sentences and normalize by the maximum number of synsets between the two sentences.

# Models and additional ideas

## Models

- SVM
- XGBoost
- Random Forest

## Additional ideas

- Oversampling SMTeuroparl and MSRpar
- Feature selection with Random Forest based on importance
- Normalizing the features
- PCA on the features



## Initial approach

- Features
  - Basic features (without Levenshtein)
  - Synset best possible pair similarity (Path and Wu-Palmer)
  - Jaccard similarities for tokens and lemmas
  - N-grams jaccard similarities
- Using XGBoost (with a grid search for hyperparameters)

# Course of Actions

- Random Forest (with a grid search for hyperparameters)

# Course of Actions

- Random Forest (with a grid search for hyperparameters)
- Manual feature selection

# Course of Actions

- Random Forest (with a grid search for hyperparameters)
- Manual feature selection
- Leacock-Chodorow similarity

# Course of Actions

- Random Forest (with a grid search for hyperparameters)
- Manual feature selection
- Leacock-Chodorow similarity
- Resnik and Jiang-Conrath similarities

# Course of Actions

- Random Forest (with a grid search for hyperparameters)
- Manual feature selection
- Leacock-Chodorow similarity
- Resnik and Jiang-Conrath similarities
- Support Vector Regression

# Course of Actions

- Random Forest (with a grid search for hyperparameters)
- Manual feature selection
- Leacock-Chodorow similarity
- Resnik and Jiang-Conrath similarities
- Support Vector Regression
- Oversampling

# Course of Actions

- Random Forest (with a grid search for hyperparameters)
- Manual feature selection
- Leacock-Chodorow similarity
- Resnik and Jiang-Conrath similarities
- Support Vector Regression
- Oversampling
- PCA



# Course of Actions

- Random Forest (with a grid search for hyperparameters)
- Manual feature selection
- Leacock-Chodorow similarity
- Resnik and Jiang-Conrath similarities
- Support Vector Regression
- Oversampling
- PCA
- Lesk Jaccard similarities

- Random Forest (with a grid search for hyperparameters)
  - Manual feature selection
  - Leacock-Chodorow similarity
  - Resnik and Jiang-Conrath similarities
  - Support Vector Regression
  - Oversampling
  - PCA
  - Lesk Jaccard similarities
- SentiWordNet

# Course of Actions

- Random Forest (with a grid search for hyperparameters)
  - Manual feature selection
  - Leacock-Chodorow similarity
  - Resnik and Jiang-Conrath similarities
  - Support Vector Regression
  - Oversampling
  - PCA
  - Lesk Jaccard similarities
- SentiWordNet
  - Considering the number of occurrences for the N-grams

# Course of Actions

- Random Forest (with a grid search for hyperparameters)
- Manual feature selection
- Leacock-Chodorow similarity
- Resnik and Jiang-Conrath similarities
- Support Vector Regression
- Oversampling
- PCA
- Lesk Jaccard similarities
- SentiWordNet
- Considering the number of occurrences for the N-grams
- Ignore stopwords in Jaccard similarities

# Course of Actions

- Random Forest (with a grid search for hyperparameters)
- Manual feature selection
- Leacock-Chodorow similarity
- Resnik and Jiang-Conrath similarities
- Support Vector Regression
- Oversampling
- PCA
- Lesk Jaccard similarities
- SentiWordNet
- Considering the number of occurrences for the N-grams
- Ignore stopwords in Jaccard similarities
- Feature selection with Random Forest Importance

# Course of Actions

- Random Forest (with a grid search for hyperparameters)
- Manual feature selection
- Leacock-Chodorow similarity
- Resnik and Jiang-Conrath similarities
- Support Vector Regression
- Oversampling
- PCA
- Lesk Jaccard similarities
- SentiWordNet
- Considering the number of occurrences for the N-grams
- Ignore stopwords in Jaccard similarities
- Feature selection with Random Forest Importance
- Removing punctuation

# Course of Actions

- Random Forest (with a grid search for hyperparameters)
- Manual feature selection
- Leacock-Chodorow similarity
- Resnik and Jiang-Conrath similarities
- Support Vector Regression
- Oversampling
- PCA
- Lesk Jaccard similarities
- SentiWordNet
- Considering the number of occurrences for the N-grams
- Ignore stopwords in Jaccard similarities
- Feature selection with Random Forest Importance
- Removing punctuation
- Lesk synset pair similarities

# Course of Actions

- Random Forest (with a grid search for hyperparameters)
- Manual feature selection
- Leacock-Chodorow similarity
- Resnik and Jiang-Conrath similarities
- Support Vector Regression
- Oversampling
- PCA
- Lesk Jaccard similarities
- SentiWordNet
- Considering the number of occurrences for the N-grams
- Ignore stopwords in Jaccard similarities
- Feature selection with Random Forest Importance
- Removing punctuation
- Lesk synset pair similarities
- Normalizing Variables



# Course of Actions

- Random Forest (with a grid search for hyperparameters)
- Manual feature selection
- Leacock-Chodorow similarity
- Resnik and Jiang-Conrath similarities
- Support Vector Regression
- Oversampling
- PCA
- Lesk Jaccard similarities
- SentiWordNet
- Considering the number of occurrences for the N-grams
- Ignore stopwords in Jaccard similarities
- Feature selection with Random Forest Importance
- Removing punctuation
- Lesk synset pair similarities
- Normalizing Variables
- Removing PCA

# Results

<b>Dataset</b>	<b>Pearson Correlation</b>
SMTeuroparl	0.5794
MSRvid	0.8375
MSRpar	0.6148
surprise.OnWN	0.7240
surprise.SMTnews	0.5576
All datasets	0.7382

**Table:** Testing Pearson Correlation on Different Datasets

# Takeaways

- Impact of the different Features
- Gap with modern techniques
- Importance of cleaning and preprocessing
- Importance of effective implementation
- Importance of the choice of the model

# Questions?