Debate Topic Expansion

Chieh Kang

Content

1. Introduction
2. Model
   a) Overview
   b) Selecting Topics & Building Corpus & Translation
   c) Pattern Extraction
   d) Filter
   e) Input Features
   f) Training
3. Experiments
4. Final model
5. Future works

Content

1. Introduction
2. Model
   a) Overview
   b) Selecting Topics & Building Corpus & Translation
   c) Pattern Extraction
   d) Filter
   e) Input Features
   f) Training
3. Experiments
4. Final model
5. Future works

Introduction

Motivation

- IBM's project debater, the first AI system that can debate on complex topics
- One of the very first steps is topic expansion
- Given a debate topic, it's important to first define and then to extend content

Introduction

Tasks

1. Extracting possible EC (expansion concept/topic)from sentences containing DC(debate concept/topic)

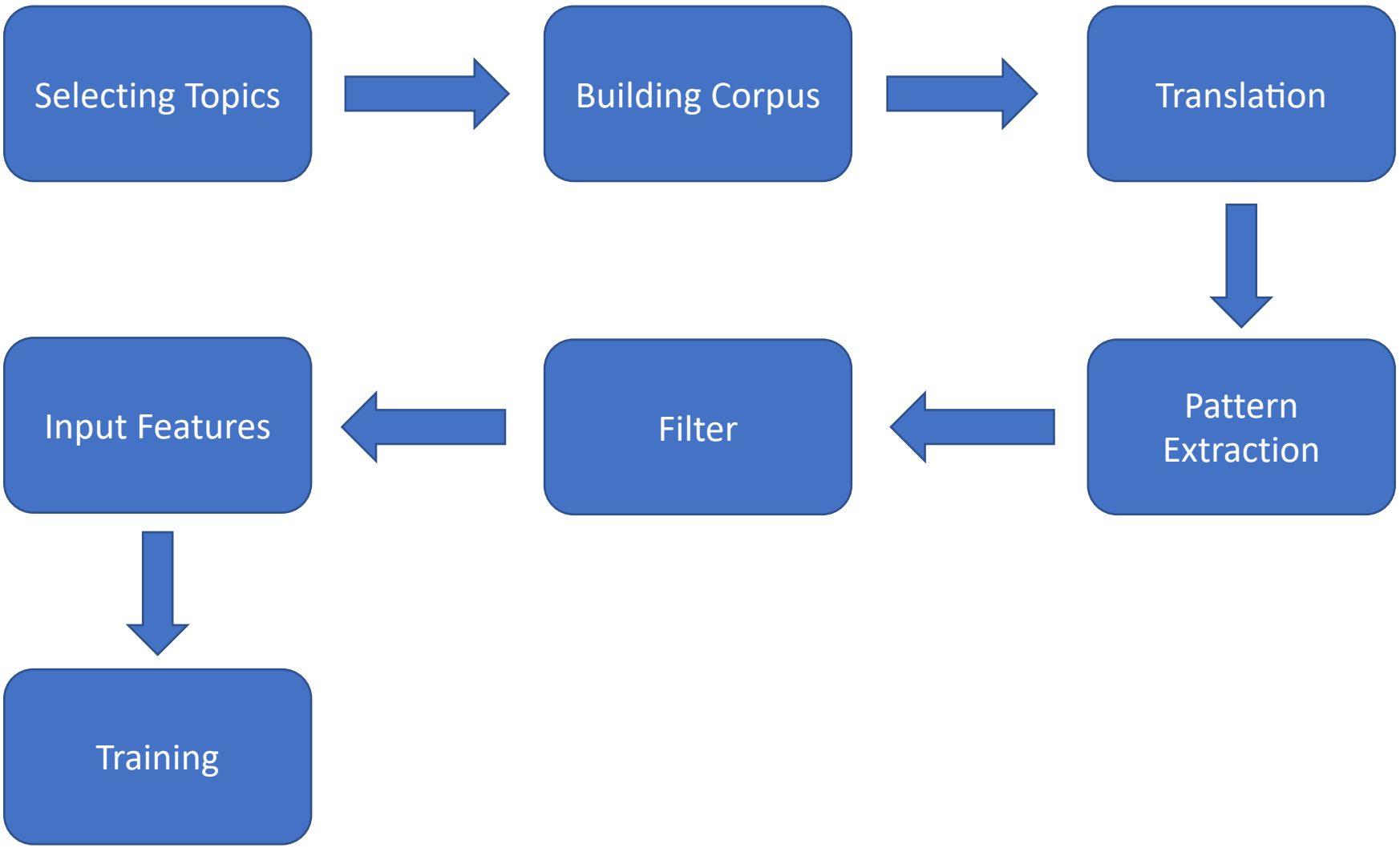
2. Debate topic expansion in German through translation

3. The goal is to build a model recognizing good and bad expansions

Content

1. Introduction
2. Model
   a) Overview
   b) Selecting Topics & building Corpus &Translation
   c) Pattern Extraction
   d) Filter
   e) Input Features
   f) Training
3. Experiments
4. Final model
5. Future works

Overview


Selecting Topics
Building Corpus
Translation
Pattern Extraction
Filter
Input Features
Training

Topics, Corpus and Translation

1. Selecting Topics:
   - 50 debate topics selected manually
   - Also wiki topics

2. Building Corpus:
   - Extract the wiki pages of the outlinks of these topics
   - About 500 million characters

3. Translation:
   - Use Libretranslate to translate into german

Pattern Extraction

Extract Pattern

1. Get sentences containing our concepts and patterns with help of regular expression

2. Use StanzaClientAPI (dependency parsing) to extract EC (detail next page)

A ist ein Beispiel für B

Covid-19 ist ein Beispiel für B

3. Delete duplicates

Pattern Extraction

Extract Pattern (steps)

1. Label the last word of the pattern as L

2. Find the head of L, labeled as H

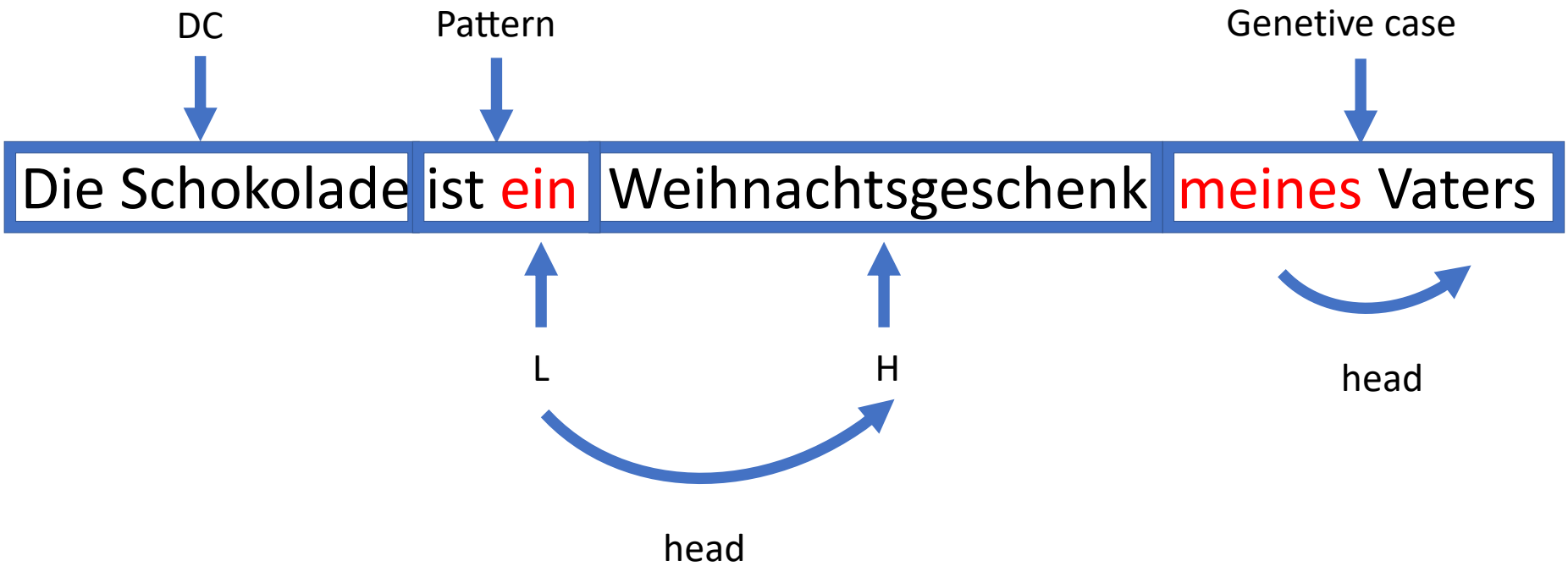
3. Check if the next word of H is a genitive case

4. If not extract the words between L and H

5. If exists, then find the head of the genitive case and extract the words from L

Pattern Extraction

Extract Pattern (example)



EC: Weihnachtsgeschenk meines Vaters

Filter

List of filters

1. Stop words
2. Substring
3. Frequency ratio (>0.01)
4. Cosine similarity (>0.2)

255 pairs left (40 good expansions, 215 bad expansions)

Imbalanced!!

Input Features

List of input features

1. cosine_similarity
2. hypernym
3. hyponym
4. co-hypernym
5. synonym
6. DC_sentiment
7. EC_sentiment
8. freq_ratio

Training

Training steps

1. Labelling
   - Manually label good and bad expansions
2. Preprocessing
   - The features remain and the target is extracted
3. Split train test data
   - Use stratified train test split and split-ratio is 0.1
4. Grid search cross validation
   - Use grid search cross validation to fine-tune the model

Note: The f1-score is calculated in macro-f1-score, since it is important for us to recognize good topic expansions

Content

1. Introduction
2. Model
   a) Overview
   b) Selecting Topics & Building Corpus & Translation
   c) Pattern Extraction
   d) Filter
   e) Input Features
   f) Training
3. Experiments
4. Final model
5. Future works

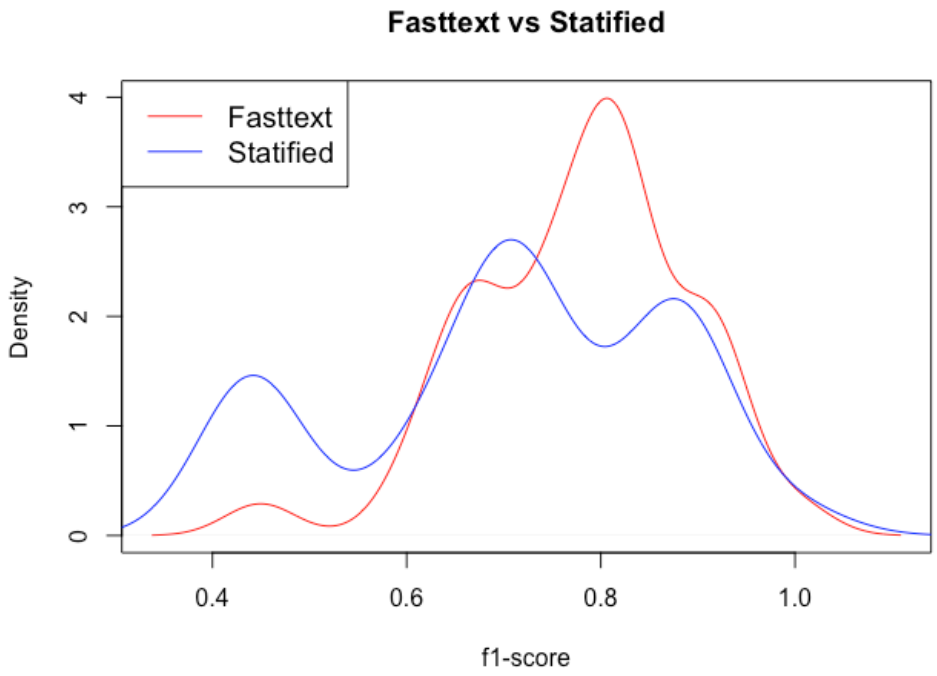
Statified contextualized vs Fasttext

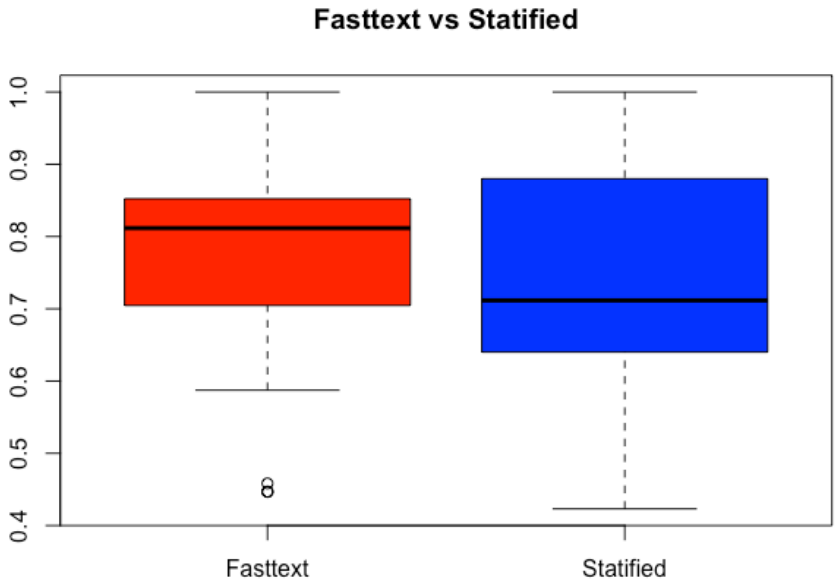
Objectives and how it's done

- Find out if the types of word embedding effect the performance
- Compare the result of fasttext and self-trained embedding (trained with whole german wiki and Bert)
- Normally the statified contextualized word embedding should output better result

Statified contextualized vs Fasttext

Result





p-value for fasttext == self-trained embedding: 2.074e-05

Statified contextualized vs Fasttext

Result

- Fasttext average: 0.7785870
- Statified contextualized average: 0.7083629

- Assumed reason: There is some word embedding that Statified contextualized do not contain, but on the other hand fasttext can handle them

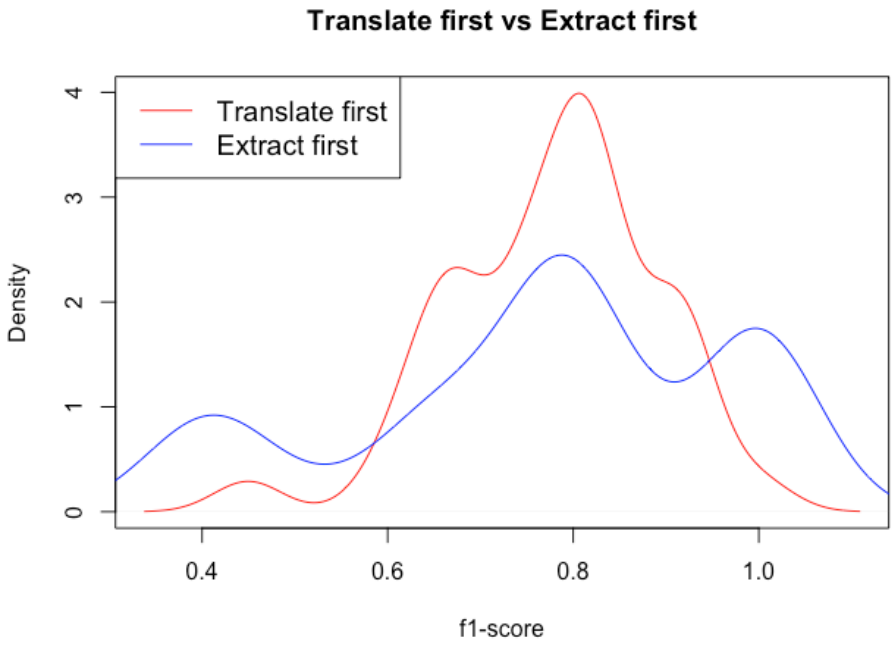
Translate or Extract first?

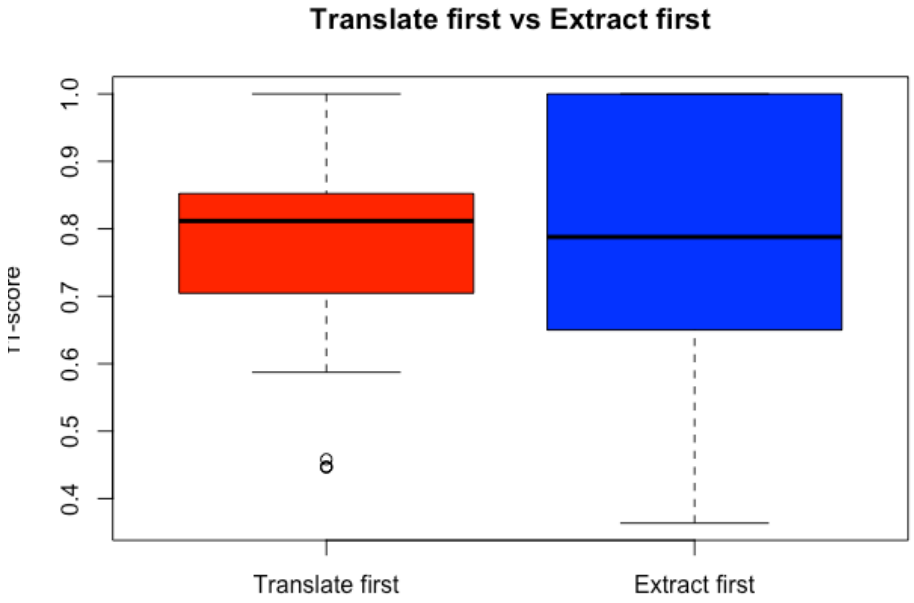
Objectives and how it's done

- Two methods:
  1. Translate corpus into german then extract the expansion topics
  2. Extract the expansion topics then translate the extracted expansion topics into german

- If ”Extractionfirst” has better performance then there is no need to translate the whole corpus and thus less computational cost

Translate or Extract first?





Translate or Extract first?

Result

- Translate first: 0.7785870
- Extract first: 0.7672677

- ”Translate first” has more than 250 pairs, whereas ”Extract first” has only a little bit more than 60 pairs

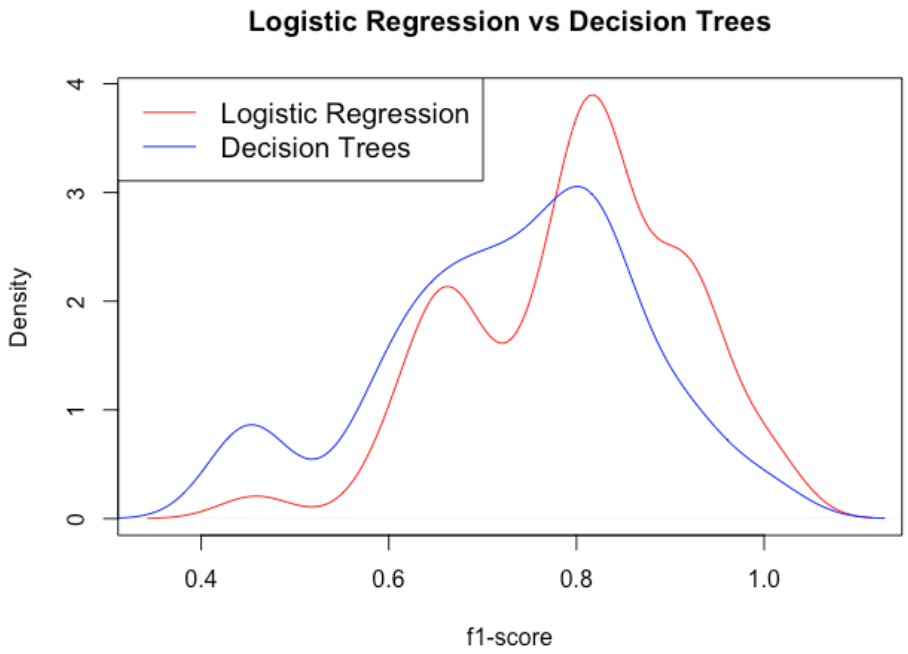
Logistic regression vs Decision trees

Objectives and how it's done

- Determine which of machine learning methods fit our data better

- Differences between two models:
  1. The ways they separate data are different. Logistic regression is a linear classifier, whereas decision trees are a non-linear classifier
  2. Decision trees have normally a higher computational cost than logistic regression.

Logistic regression vs Decision trees



Decision trees: 0.7310355

Logistic: 0.7974715

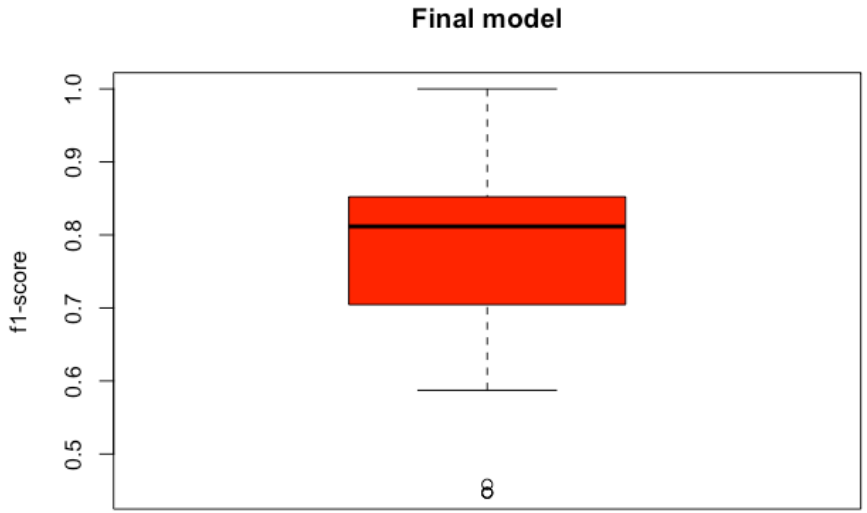
Content

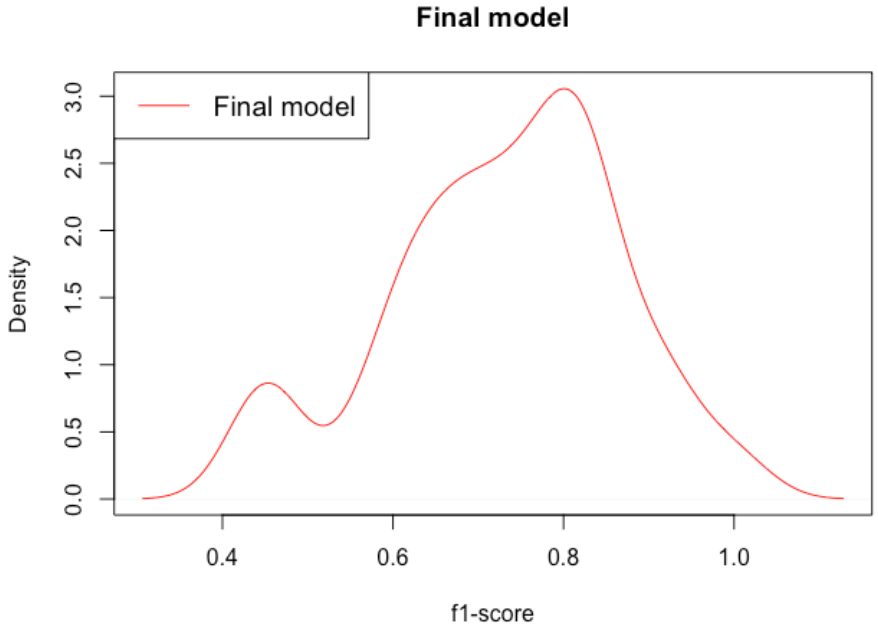
1. Introduction
2. Model
   a) Overview
   b) Selecting Topics & Building Corpus & Translation
   c) Pattern Extraction
   d) Filter
   e) Input Features
   f) Training
3. Experiments
4. Final model
5. Future works

Final model

	F1-score
Fasttext	0.7785870
Statified contextualized	0.7083629
Translate first	0.7785870
Extract first	0.7672677
Logistic regression	0.7974715
Decision trees	0.7310355

Final model





Final f1-score: 0.7974715

Content

1. Introduction
2. Model
   a) Overview
   b) Selecting Topics & Building Corpus & Translation
   c) Pattern Extraction
   d) Filter
   e) Input Features
   f) Training
3. Experiments
4. Final model
5. Future works

Future works

- Corpus size-translation tradeoff
- Word embedding: How to deal with words out of vocabulary
- Filter and feature selection

Thank you