InfoQubes

Syntax-Based Sentiment Analysis of Tweets in Russian

Yu. Adaskina, P. Panicheva, A. Popov

InfoQubes

Dialogue 2015, Moscow, May 27-30, 2015

Introduction

SentiRuEval

- Independent Sentiment analysis evaluation;
- 2015’s evaluation includes a track on three-way classification of tweets about banks and telecommunications companies;

Our Approach

- We performed full morpho-syntactic analysis of the data using InfoQubes parser;

- With normalized forms and syntactic relations as features, we applied SVM classification.

- We tested various feature sets as parameters;
- We also tried Naïve Bayes, which showed slightly lower results than SVM.

Related Work

Pang, Lee, Vaithyanathan 2002

- Is widely considered the main work on ML classification methods for sentiment analysis;

- Further research has been done to establish the ultimate feature set for each specific task, among these are:
  - word forms;
  - normalized words;
  - phrases;
  - n-grams;
  - binary occurrences;
  - syntactic relations.

Syntactic features

- Syntactic analysis in NLP systems is
  - Time-consuming;
  - Expensive.

- However, see Furnkranz, Mitchell, Rilof 1998, Caropreso, Matwin, Sebastiani 2001, Nastase, Shirabad, Caropreso 2006, Matsuko et al. 2005, Bethard, Martin 2007, Zhang et al. 2007, Zhao, Grishman 2005 among others for their use of syntactic information for ML classification.

Matsumoto et al. 2005

- Deal with sentiment classification based on syntactic relations;
- They parsed frequent sub-trees;
  - We only used syntactically related word pairs;
- They relied on syntactic trees only;
  - We combined syntactic features with other.

The Task

The dataset (see Loukachevitch et al. 2015 for details)

Training set	**5000 tweets on 8 banks, manually annotated by SentiRuEval experts**	**5000 tweets on 7 telecom companies, manually annotated by SentiRuEval experts**
Evaluation set	**5000 tweets on 8 banks**	**5000 tweets on 7 telecom companies**

- The test set had been annotated with neutral values for every company that was mentioned in the tweet;
- The participants needed to perform automatic sentiment analysis on the test set, which is either to retain a neutral annotation for the appropriate brand, or to change it to negative annotation or to a positive one;
- The evaluation set was annotated by three assessors, and tweets where there was no agreement between the experts (at least two of the three), were excluded.

Our Method

Support Vector Classifier

- Linear SVC implementation from Python sklearn/scikit-learn library;
- We extracted the following types of features from tweets:
  - lemmatized unigrams;
  - lemmatized bigrams;
  - binary syntax relation between lemmatized unigrams;
  - + optional negation marker for each feature type;
- During SVC learning procedure we combined different feature types to achieve higher results.

InfoQubes platform

- Parses texts
    - Mines lemmas;
    - Assigns morphological tags;
    - Produces syntactic trees.

Syntactic relation as feature is a combination of:

- Source word;
- Target word;
- Relation type.

- Our system features 16 syntactic relation types, one of which has 4 subtypes; we treat those as different relation types in our parametric model.

Feature type combinations

- Used combinations:
    - Lemmas;
    - Relations;
    - Bigrams;
    - Lemmas + relations;
    - Bigrams + relations;
    - Bigrams + lemmas;
    - Bigrams + lemmas + relations;
- Options:
    - Negation marker;
    - Brand-containing feature exclusion.

Classification and scoring

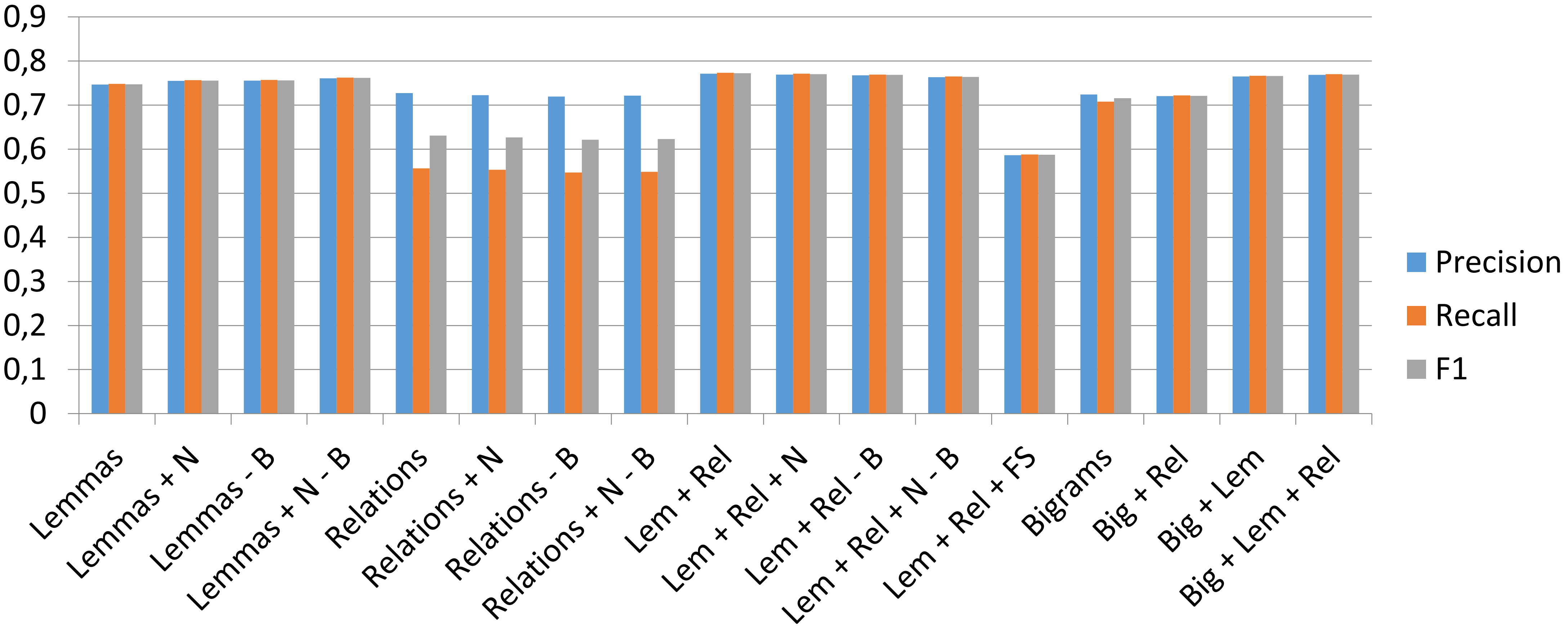
- For test and reference sets a list of the following triplets is compiled: document_id|brand_id|sentiment_score;
- Recall, precision and Micro F1-measure is computed over the sets of triplets obtained from SVM classifier and manually annotated documents.

Preliminary Experiments

- We conducted some preliminary experiments applying ten-fold cross-validation to the training dataset.

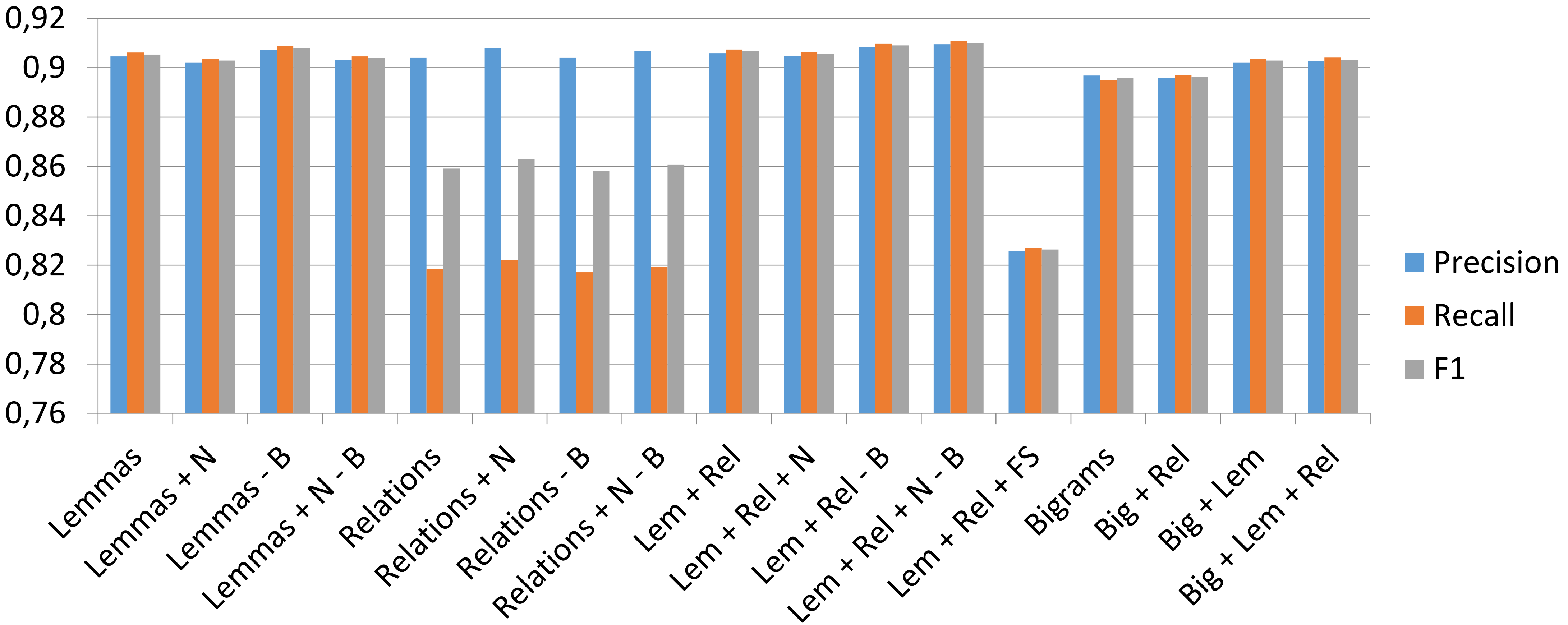
InfoQubes

Preliminary Telecom results



InfoQubes

Preliminary Banks results



Results of preliminary experiments

- A combination of lemmas and syntax relations provides the best results for both banks and telecom companies;
- Negation and brand name removal options do not considerably affect the performance;
- Bigrams and lemmas work slightly worse than relations and lemmas;
- Naïve Bayes classification has confirmed all these tendencies with a small decrease in performance;
- Feature selection significantly reduces performance; this is probably due to the high sparsity of data.

Final Experiment

Final experiment feature types

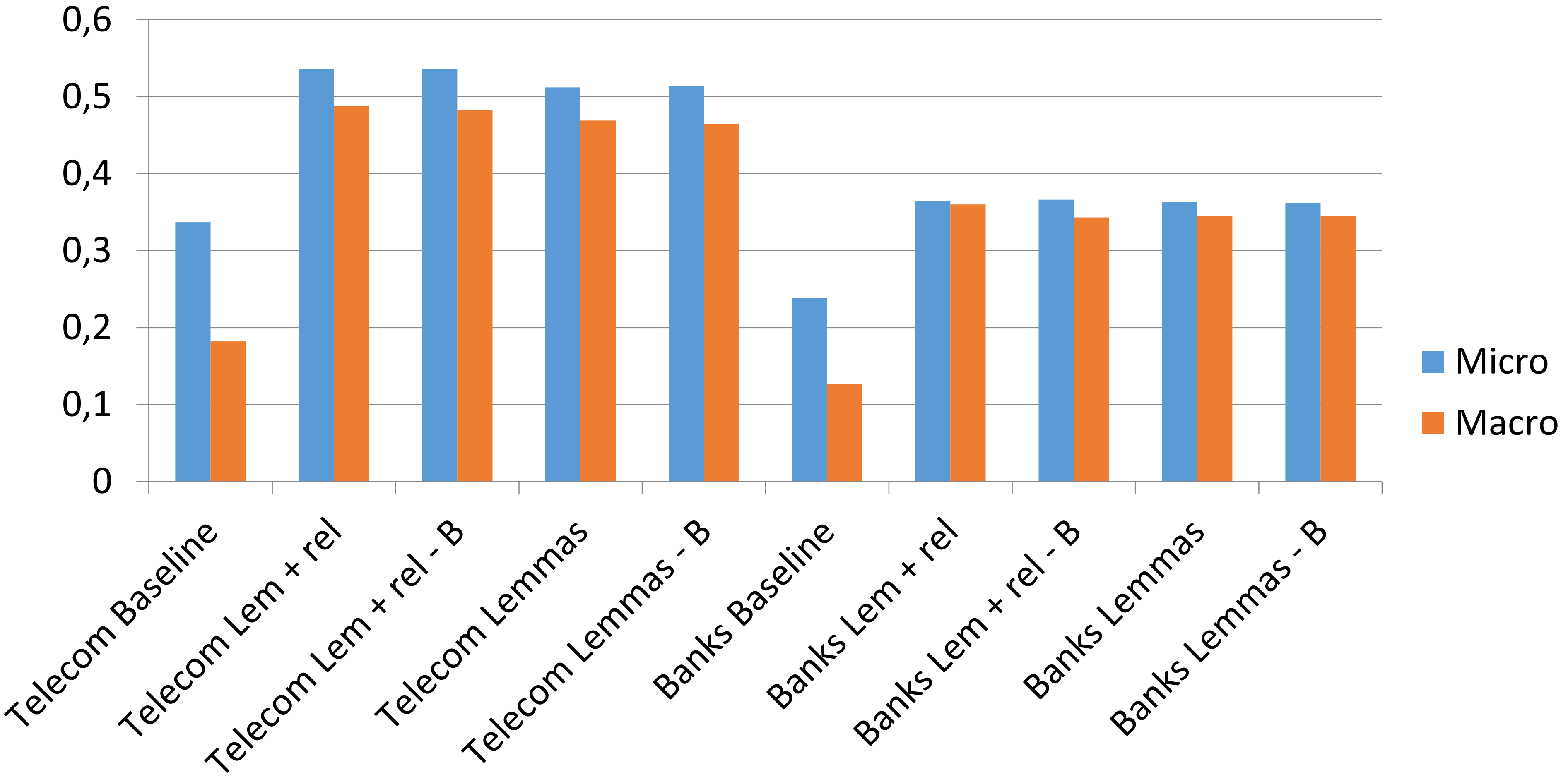
- Lemmas + relations;
- Lemmas + relations with brand names removed;
- Bigrams;
- Lemmas*;
- Lemmas with brand names removed*.

** Out of competition results*

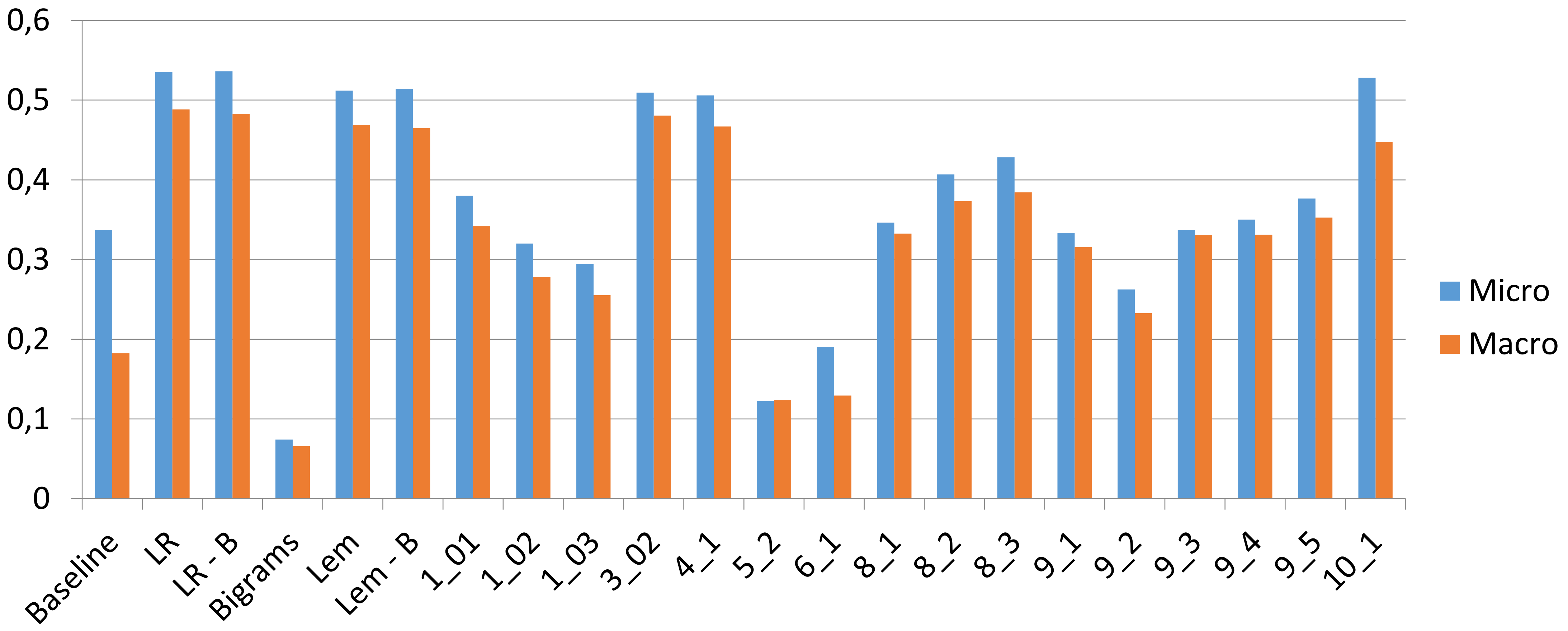
SentiRuEval evaluation

- SentiRuEval approach differed from ours:
  - We compute Micro F-measure over all three classes;
  - SentiRuEval computes Micro and Macro F-measures over positive and negative classes only;

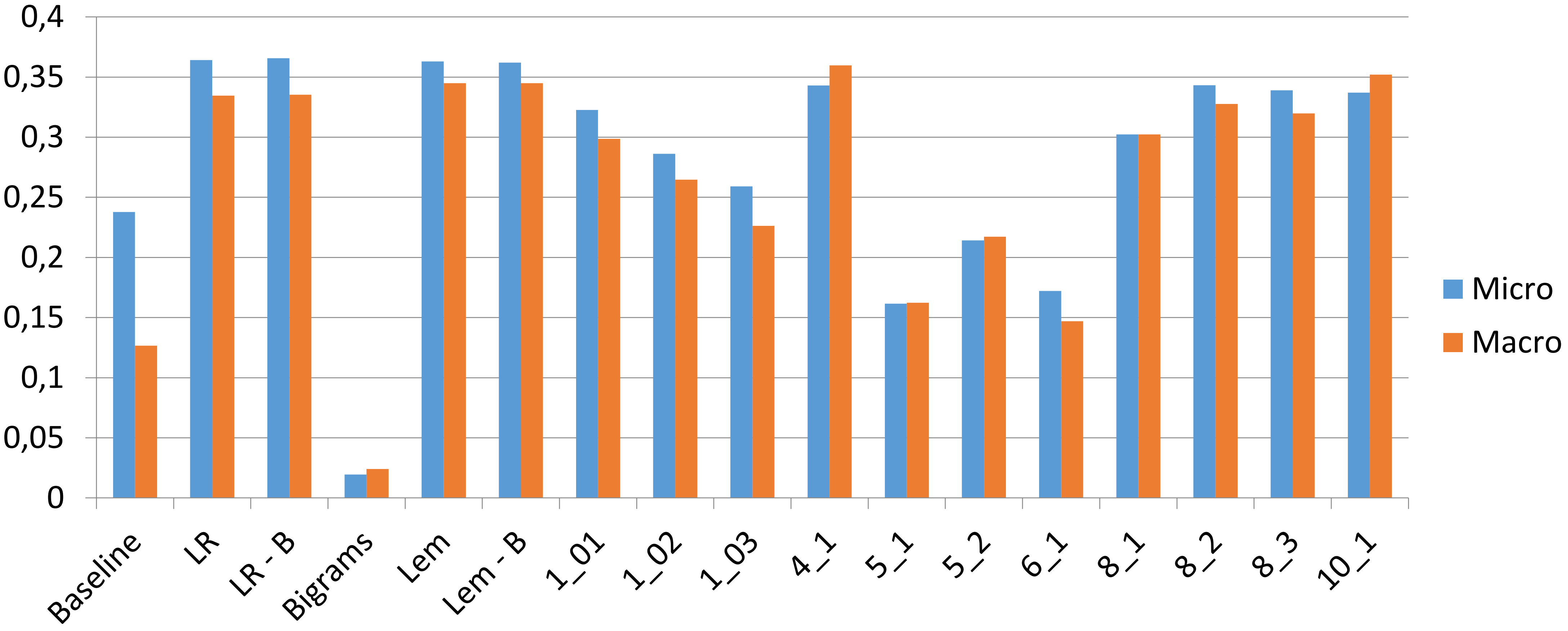
Our SentiRuEval results



Overall SentiRuEval Telecom results



Overall SentiRuEval Banks results



Summary

- We performed a three-way classification of tweets using SVC;
- We used different feature combinations;
- It seems that syntactic features increase overall performance.

Conclusion

Result evaluation

- We have applied our method for two different topics yielding high performance in both of them;
- We included syntactic information as features for ML:
  - In general it improved performance;
  - No decrease in performance has been detected;
- We mined our features using our morphosyntactic parser, which has been successfully used for another semantic tasks.

Data sets evaluation

- Our data sets has proven to be quite sparse and modest-sized:
  - SVM classifier appears to be the best choice;
  - Negation or brand-name semantics do not affect the performance much, with syntactic relations possibly conveying most of the information carried by the negation option;
  - Most of the complex features had low frequencies;
  - Any feature selection or filtering we performed proved to be quite ineffective due to sparseness and resulting in high decrease in performance, this could be an option if we boost feature occurrence by, for example, substituting words with semantic classes.

Thank you!