RULE-BASED APPROACH TO SENTIMENT ANALYSIS AT ROMIP 2011

Dmitry Kan (dmitry.kan@gmail.com)

AlphaSense Inc

This paper describes rule-based approach to sentiment analysis, that aims at shallow parsing of an input text in the Russian language and applying a set of linguistic rules for resolving a sentiment of a given chunk (subclause, sentence or text). The algorithm shows decent perfomance (90% precision for positive class) for the cases when annotators agreed on a sentiment label and has the feature of the text object related sentiment classification.

Key words: sentiment classification, sentiment analysis, opinion mining, ROMIP, blog data, rule-based sentiment analysis

1. Introduction

Sentiment analysis while being a subtask of artificial intelligence (or more strictly, Natural Language Processing) remains rather vagualy defined. This is supported by a cross-annotator agreement levels, which maximize at around 80% [1]. This can in a way be considered as a target level for sentiment detection accuracy, starting from which further improvements require more fine-grained tuning of algorithms. Partially, complexity of the task definition refers to how each human annotator perceives a sentiment expressed in an utterance depending on his/her own current mood, personal attitude to utterance's subject, annotation's goal defined. Rule-based algorithms bring a good level of clarity which levereges the ability to answer simple question: why did the system classified given chunk to a certain sentiment class? The paper is organized as follows. Section 2 presents the rule-based algorithm for the Russian language that can classify to 2 (negative and positive) and 3 (negative, neutral and positive) classes. In Section 3 we give a break down by each metrics in relation to other participant systems. Section 4 concludes the paper and outlines open problems to be solved.

2. Rule-based linguistic algorithm

The aim of this approach is shallow parsing and is twofold: high speed of processing and flexibility to unknown, potentintially, spoken utterances. It is worth to mention, that for rule-based systems the complexity of sentiment detection task rougly grows as follows: (easiest) sentiment of a subclause, (moderate) sentiment of a sentence, (hardest) sentiment of the entire text.

In our implementation described further, object oriented and general level sentiment detection algorithm branches have scored about same, which shows high level of inter-consistency between the branches. In the listing 1 we present a pseudo code of the rule-based sentiment algorithm, which classifies a given chunk to a sentiment on a general level (i. e. without relation to any object).

```
func classifySentence(sentence)
{
  positiveScore = 0
  negativeScore = 0
  totalScore = 0
  NEGATION_WEIGHT = 2
  OPPOSITE_CONJUNCTION_WEIGHT = -1
  Let NEGATIONS be set of negation words
  Let P be set of positive words
  Let N be set of negative words
  Let O be set of opposite conjunctions
  words[] = getWords(sentence) // returns list of words
in base form
  // 1. analyze negation word window first
  // negation word examples: не, ни, еле, нет
  i = 0
  // iterate over words
  do
    if word_i ∈ NEGATIONS then
      for each word ∈ [word_i+1, word_i+3] do
        if word ∈ P then
          positiveScore = positiveScore — NEGATION_WEIGHT
        end if
        if word ∈ P then
          negativeScore = negativeScore — NEGATION_WEIGHT
        end if
      end for
    end if
    i = i + 1
  while i<|words|-3
  // 2. Calculate sentiment rank of separate words — second
pass
  i = 0
  do
    if word ∈ P then
      positiveScore = positiveScore + 1
    end if
    if word ∈ N then
      negativeScore = negativeScore + 1
```

```
    end if
    i = i + 1
  while i<|words|
  // 3. Third pass is: global view of a sentence for han-
dling opposite conjunctions,
  // example of which are: а, но, хотя
  i = 0
  sentimentCount = 0
  oppositeSentimentScore = 0
  do
    if word ∈ P then
      sentimentCount = sentimentCount + 1
    end if
    if word ∈ O then
      sentimentCount = sentimentCount + 1
    end if
    if word ∈ O then
      oppositeSentimentScore = OPPOSITE_CONJUNCTION_WEIGHT
* sentimentCount / 2
    end if
    i = i + 1
  while i<|words|
  totalScore = positiveScore – negativeScore +
oppositeSentimentScore
  if totalScore > 0 then
    sentiment is positive
  else if totalScore < 0
    sentiment is negative
  else
    sentiment is neutral
}
```

Listing 1. Pseudo-code of the rule-based sentiment classification algorithm for Russian

The algorithm has three main components:

- Polarity dictionaries for Russian;
- Set of negations of Russian, that tend to noticeably affect on polarity of connected word(s): ***не*** плохо (***not*** bad); also gap between words are processed correctly, for example: Я ***не*** сильно ***люблю*** это (I ***do not*** strongly ***like*** this);
- Set of opposite conjunctions of Russian, which affect on polarity of sentence's subclauses in relation to each other: Большинству это всё нравится, ***а*** мне нет (Majority likes this, ***but*** I do not).

The object related sentiment detection is done in the following way:

- First each sentence of the input text is examined for the presense of the keywords of the object;
- If the sentence was found, it is checked for the presence of conjuctions or other boundaries of subclauses (like punctuation);
- If there is no boundary found, the sentiment of the entire found sentence is detected according to the algorithm described above;
- If there is a boundary, the subclause containing the keywords is identified and sentiment of the subclause is detected according to the algorithm described above.

Polarity information of separate words is contained in the corresponding dictionaries. Each word is stored in its base form (lemma). Each word in a polarity dictionary was selected based on its unambigous polarity value. That means, that if a word had different polarities in different contexts, it would not be included into the dictionary. Here is an excerpt from both the positive and the negative dictionaries:

Positive polarity (1739 entries in total currently):
благо (good, welfare)
благой (good)
вкусный (delicious)
грандиозный (immense)
…
классный (cool)
красивый (beautiful)
красота (beauty)
…
рекомендовать (recommend)
…
симпатичный (attractive)
скидка (discount)

Negative polarity (2338 entries in total currently):
бездарность (lack of talent)
бесить (enrage)
болеть (be sick)
больница (hospital)
больно (it is painful)
…
грустить (be sad)
дождь (rain)
дорого (expensive)
…
жесткий (hard, rude)
жесть (nasty)

жуткий (terrible, weird)
…
подделка (fake)
подстава (bumper crime)
покраснеть (turn red)
поломаться (break)
…
скучный (boring)

When doing a lookup in the polarity dictionary, first word's lemma is calculated using lemmatizer for Russian (based on Zaliznyak's grammatical dictionary [7]). Then polarity of a given word changes to opposite, if a negation was found in front of it (including gaps between negation and the word). When processing conjunctions, algorithm goes to the level of subclauses and a sentiment of one subclause affects on sentiment of another subclause, joined with a conjunction. It is evident from the algorithm's psedo-code, that it classifies a chunk to 3 classes (negative, neutral and positive). For the 2 class sentiment classification task, we used classifier ensemble with the statistical method (multinomial Naïve Bayes classifier [6], which was slightly transformed and trained on lemmatized unigrams and bigrams) described in [3]. In this case 311 neutrally annotated chunks out of 50 177 chunks were passed down to the statistical classifier for the 2 class annotation. The 3 class classification task did not have neutral class as such, but rather a class that signifies sufficient presence of positive and negative polarity in a given chunk. In this case we used neutral class, because when positive and negative scores are equal in absolute values and there is no opposite conjunctions found, the final sentiment score is zero (neutral class). For the 3 class task we used the rule-based classifier only.

3. Evaluation

For evaluating methods of 2 class task the following scores were calculated by ROMIP organizers [2]:

$$P = \frac{P_N + P_P}{2}, R = \frac{R_N + R_P}{2}, F = \frac{F_N + F_P}{2},$$

where precision, recall and F-metric for a positive class are:

$$P_p = \frac{tp}{tp + fp}, R_p = \frac{tp}{tp + fn}, F_p = 2 * \frac{P_P * R_P}{P_P + R_P}, \text{where}$$

tp — number of true positives, fp — number of false positives, fn — number of false negatives.

For negative class we have:

$$P_N = \frac{tn}{tn + fn}, R_N = \frac{tn}{tn + fp}, F_p = 2 * \frac{P_N * R_N}{P_N + R_N}, \text{where}$$

tn — number of true negatives.

Overall accuracy of a method then is:

$$A = \frac{tp + tn}{tp + tn + fp + fn}$$

For evaluating methods of the 3 class task the following scores were used by ROMIP organizers [2]:

$$P_x = \frac{tp_x}{tp_x + fp_x}, R_x = \frac{tp_x}{tp_x + fn_x}$$ — precision and recall for class x,

then macro-statistics are defined as follows:

$$P_{macro} = \frac{1}{|S|}\sum_{x \in s} P_x, R_{macro} = \frac{1}{|S|}\sum_{x \in s} R_x, F_{macro} = 2 * \frac{P_{macro} * R_{macro}}{P_{macro} + R_{macro}}$$

From three evaluation topics (books, films and digital cameras) the proposed system has performed best in ensemble with [3] for films: scoring 14th out of 27 participants for 2 class task and alone scoring 14th out of 21 participants for 3 class task. We should mention that the rule-based system was not specifically tuned to the topic neither on the level of rules, nor on the polarity dictionaries. Table 1 shows system performance (classifier ensemble denoted as **ssee** and rule-based classifier denoted as **sse**) for the 2 class sentiment classification in relation to other closest from the top participant systems as well as average over scores of all participant systems.

Table 1. Official ROMIP 2011 results for 2 class sentiment classification track for films topic (first 16 out of 27 total runs by participants sorted by F_MEASURE_AND F with average over all runs)

	P	R	F	A	P_p	R_p	F_p	P_N	R_N	F_N
xxx-23	0.7596	0.7813	0.7696	0.8750	0.9344	0.9167	0.9254	0.5849	0.6458	0.6139
xxx-9	0.6798	0.7718	0.7021	0.8013	0.9430	0.8144	0.8740	0.4167	0.7292	0.5303
xxx-13	0.6693	0.7045	0.6832	0.8173	0.9124	0.8674	0.8893	0.4262	0.5417	0.4771
xxx-17	0.6611	0.7453	0.6799	0.7853	0.9339	0.8030	0.8635	0.3882	0.6875	0.4962
xxx-3	0.6619	0.7074	0.6780	0.8077	0.9146	0.8523	0.8824	0.4091	0.5625	0.4737
xxx-12	0.6564	0.7528	0.6718	0.7692	0.9404	0.7765	0.8506	0.3723	0.7292	0.4930
xxx-18	0.6484	0.7292	0.6644	0.7724	0.9289	0.7917	0.8548	0.3678	0.6667	0.4741
xxx-4	0.6390	0.7405	0.6438	0.7340	0.9415	0.7311	0.8230	0.3364	0.7500	0.4645
xxx-10	0.6323	0.7405	0.6253	0.7051	0.9479	0.6894	0.7982	0.3167	0.7917	0.4524
xxx-26	0.6227	0.7301	0.6018	0.6731	0.9500	0.6477	0.7703	0.2955	0.8125	0.4333
xxx-14	0.7154	0.5805	0.5996	0.8526	0.8682	0.9735	0.9179	0.5625	0.1875	0.2813
xxx-5	0.6119	0.7064	0.5961	0.6763	0.9358	0.6629	0.7761	0.2880	0.7500	0.4162
xxx-6	0.6117	0.5701	0.5805	0.8205	0.8662	0.9318	0.8978	0.3571	0.2083	0.2632
seee	**0.5852**	**0.6572**	**0.5641**	**0.6506**	**0.9144**	**0.6477**	**0.7583**	**0.2560**	**0.6667**	**0.3699**
sse	**0.5845**	**0.6581**	**0.5574**	**0.6378**	**0.9171**	**0.6288**	**0.7461**	**0.2519**	**0.6875**	**0.3687**
avg	0.6006	0.6340	0.5596	0.6926	0.9120	0.7165	0.7787	0.2891	0.5205	0.3406

Table 2 shows performance of rule-based classifier (denoted as **sse**) the for 3 class sentiment classification in relation to other closest from the top participant systems as well as average over scores of all participant systems. This time the average (denoted avg) is above the system performance.

Table 2. Official ROMIP 2011 results for 3 class sentiment classification track for films topic (first 15 out of 21 total runs by participants sorted by F_MEASURE_AND F with average over all runs)

	P	R	F	A
xxx-10	0.6037	0.4745	0.5302	0.7338
xxx-19	0.5982	0.4715	0.5272	0.7338
xxx-9	0.5989	0.4720	0.5271	0.7338
xxx-1	0.5995	0.4704	0.5271	0.7300
xxx-6	0.6060	0.4388	0.4454	0.7034
xxx-20	0.5934	0.4343	0.4424	0.6996
xxx-11	0.5738	0.4260	0.4399	0.6996
xxx-12	0.5394	0.4110	0.4252	0.6692
xxx-18	0.5035	0.3532	0.4121	0.5894
avg	0.4575	0.3565	0.3469	0.5633
xxx-0	0.3521	0.3353	0.3239	0.6882
xxx-8	0.3521	0.3211	0.3226	0.5513
xxx-2	0.4712	0.3229	0.3207	0.5475
sse	**0.4945**	**0.2746**	**0.3151**	**0.4297**
xxx-7	0.4760	0.3259	0.2846	0.6806

4. Conclusions and open problems

We have presented a rule-based sentiment detection algorithm for Russian which largely reminds approach described in [4]. The main distinctive features of the presented approach are incorporation of the language opposite conjunctions and object oriented sentiment detection. Algorithm's performance grows as more polarity words are mined for a specific topic. In this regard, exploring the methods for automatic mining of polarity dictionaries for a given domain like [5] is important for increasing the algorithm accuracy.

References

1. *Bermingham, A. and Smeaton, A.F. (2009). A study of interannotator agreement for opinion retrieval. In SIGIR, 784–785.*
2. *Chetviorkin I., Braslavskiy P., Loukachevich N. (2012). Sentiment Analysis Track at ROMIP 2011. In Dialog.*
3. *Poroshin V. (2012). Proof of concept statistical sentiment classification at ROMIP 2011. In Dialog.*
4. *Minqing Hu, Bing Liu. (2004). Mining and summarizing customer reviews. In Proc. of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining.*
5. *Chetverkin I., Loukachevitch N. (2010). Automatic Extraction of Domain-specific Opinion Words. Dialogue.*
6. *C. D. Manning and H. Schutze. Foundations of statistical natural language processing. MIT Press, 1999.*
7. *Andrey Zaliznyak. Grammaticheskij slovar' russkogo jazyka. Moskva, 1977, (further editions are 1980, 1987, 2003).*