Building a System for Stock News Generation in Russian

Liubov Nesterenko
National Research University Higher School of Economics, Moscow
lyu.klimenchenko@gmail.com

Abstract

In this paper we present an implementation of an NLG system that serves for stock news generation. The system has two modules: analysis module and NLG module. The first one is intended for processing the data on stock index changes, the second one for the news texts generation using the template-based NLG approach. The evaluation shown that both modules give relatively accurate results and the system can be used as a newsbot for stock news generation.

1 Introduction

The subject of this paper is related to the natural language generation of the stock news.

The stock quotes change considerably during the day, which means that financial media should react quickly to each remarkable change and announce the news very often and very fast. That is why it becomes necessary to make a system that receives the information about latest changes and generates short news on the base of it. In 1983 there was an attempt made by K. Kukich for financial news generation in English (Kukich, 1983). In our paper we present an NLG system for stock news generation in Russian created from scratch, it generates the news having as input the daily data on stock indexes changes.

Our goal is to describe the process of the system implementation and to discuss the problems we had to deal with. Here are the tasks we used to solve as we were working on the system development: choosing methods for index changes analysis, examining the features of financial news texts, collecting financial lexicon, choosing an NLG approach, writing a Python program and evaluating the results. The NLG component uses template-based approach. One of the reasons for that was the lack of an appropriate target corpus of stock news that could make statistical approach, like in (Sutskever et al., 2011), possible. Sometimes the template-based approach is being underestimated but some researches consider that it can be as good as the standard approach (Van Deemter et al., 2005). It could be even combined with some statistical approach, as it was done in (Howald et al., 2013). Moreover, the stock news texts have very clear and simple structure and content, that is why the template-based approach works here quite well.

2 Preliminary work

Our first step was to make some research on index changes and index behavior in general. After that we consulted with the experts on stock news and defined the types of the news that the program should generate. We decided that there are two types of news needed: the morning news and the evening news and that they should be about the changes of two main Russian indexes: MOEX (Moscow Exchange) and RTSI (Russian Trading System Index). Then we explored some financial media resources in order to understand what stock news are like and what features they have. The majority of the stock news on Russian financial media resources appeared to be whether too short (one sentence) or too long (contain lots of information regarding predictions of index future behavior and discussions). That induced us to create our own text layouts for rather short but informative news.

The content and the structure of the news texts are as follows:

Morning News

1. Trades beginning: indexs performance during the first minutes after trades started (general tendency)
2. MOEX index behaviour.

Proceedings of the 2nd International Workshop on Natural Language Generation and the Semantic Web (WebNLG), pages 37–40,
Edinburgh, Scotland, September 6th, 2016. ©2016 Association for Computational Linguistics

3. RTSI index behaviour.
4. The characteristic of the last trading day (generally and particular).
5. MOEX index behaviour yesterday.
6. RTSI index behaviour yesterday.
7. The trades value for the previous trading day.

Evening news
1. General tendency during the day.
2. MOEX index behaviour.
3. RTSI index behaviour.
4. MOEX final change.
5. RTSI final change.
6. The trades value for the day.

These text structures serve as layouts for the news. For each position one should create the suitable templates that will build up the text.

3 Methodology

The system has two components: the analysis module and the NLG module. The first one gets the daily data on indexes as input and determines the index changes using the algorithm we developed for that purpose. As output it gives the so called 'events' ('event' = index changes during the day), e.g., 'no significant changes', 'fluctuations', 'increase' etc., or the tendencies (e.g., index behavior during the first hour after the trades opening). After that the NLG module takes the event as input and generates a text according to the changes the indexes had.

In the next two subsections we illustrate how these modules function.

3.1 Analysis module

The analysis module uses the data on indexes values to detect the behavior the indexes had during one particular period of time. For the morning news generation we take the previous trading day data and the data on the first hour of the trades, for the evening news — the current day data. As it was meantioned before the stock news generation in our case demands detecting tendencies and events. Determining tendencies is retalively easy, one compares the difference between two index values: the opening value and the last value. The tendencies could be for example 'increase', 'decrease' etc. Determining events is a more complicated procedure, also the events can have a more complex structure than the tendencies. We distinguish simple events like 'increase', 'decrease' 'fluctuations' and also such events as 'significant increase & no changes' or 'increase & insignificant decrease' and other similar compound events. For events determination we use the following algorithm:

1. *Check if the index fluctuated.*

We check it by calculating the *Adjusted* R^2 value, if it appears to be less than 0.5 then we claim that the index has fluctuated.

2. *Check if there are intervals without any significant changes.*

For each interval $[t_1,\ t_2]$ we calculate the evaluative function by using this formula:

$$E = \alpha(t_1 - t_2) + \frac{\beta}{\sigma^2_{1,2} + 10^{-4}}\ ,$$

where $\sigma_{1,2}$ — standard deviation in the interval, α, β — coefficients. Thus, the bigger the interval is, the more is the evaluative function value and the less the index fluctuates, the more is the evaluative function value.

3. *Fit the polynomials of degrees 2 and 3 to the index values data.*

4. *Choose the best approximation.*

Since we had detected the interval with no significant changes and fit the polynomials, we choose the approximation that best of all corresponds to the index behavior. If the interval with no significant changes is between 1/3 and 1/2 of the whole trading day length and it is located in the first or the second half of the day, then we choose this approximation. Otherwise, we should choose between the two polynomials. Most of the cases are well described with the help of quadratic polynomial, so we have to determine if the cubic polynomial is needed or not. To acomplish that one should find the inflection point of the cubic polynomial and the difference between the *Adjusted* R^2 values of the polynomials.

5. *Apply the rules to determine the event.*

If the check for the intervals with no significant changes was positive, then the resulting 'event' will consist of 'no significant change' part and 'increase/decrease etc.' part, e.g., 'no significant change & significant decrease' or 'increase & no significant change'. If the quadratic polynomial was chosen as a suitable approximation, then one will need such parameters as the sign of the x^2 coefficient, the vertex location on the time axis and the threshold crossing (if the changes exceeded 2% relative to the opening then we call such changes significant and it affects the event) to determine the event. If the cubic polynomial was chosen as a suitable approximation, then one will need to know the the sign of the x^3 coefficient to determine the event.

By the means of this algorithm one can determine different types of index changes both simple like 'decrease', 'increase', 'fluctuations' and compound changes like 'no significant changes & increase'. The information about the index changes is further used by the NLG module for news generation.

3.2 NLG module

In this section we describe the NLG process in our system. This module uses text layouts, rules, sentence templates and financial lexicon that was collected during the work with media texts. In the result we get short texts like this one below.

(1)
Russian
Segodnya torgi prohodili v krasnoy zone. Utrom indeks MMVB nachal torgi ponizheniem i prodolzhal ustremlyat'sya vniz. V to zhe vremya, ruhnuv utrom, indeks RTS prodolzhil sil'noe padenie. Tak indeks MMVB ponizilsya na 0.39% do otmetki v 1748 punktov, a indeks RTS — snizilsya na 4.4% i dostig 804 punktov. Ob"em torgov po itogam dnya sostavil 700 millionov dollarov SSHA.

English translation
Today the trades ran in the red zone. In the morning MOEX index started to reduce and continued its lowering. At the same time RTSI fell and proceeded to decrease. So MOEX lost 0.39 % and made up 1748 points, RTSI fell by 4.4 % and reached the grade of 804 points. The volume of the trading section was 700 millions of dollars.

First of all the program takes an appropriate text layout (morning/evening). In traditional descriptions of NLG architecture, as in (Reiter et al., 2000) or (Martin and Jurafsky, 2000), one of the steps in the implementation is the macro planning. In some NLG the systems it is done authomatically but in our system the macro planning appears to be predetermined. Then the program fills in the positions with different sentence templates. For each position there are more than one suitable template. The templates are clauses with some constituents missing. Some of them have more slots, some of them less, depending on how much variance is needed. Most of the templates are independent clauses, but some of them turn out to be the constituents of one compound sentence in the result. Here are some examples of the templates the program uses for generation.

(2) [timeExpr] [subject] began to [predicate].

(3) [timeExpr] MOEX index [predicate] [value]% to [value] points, RTSI index [predicate] [value]% to [value] points.

(4) a. [timeExpr] MOEX index [predicate] [value] % to [value] points.
b. , [link] RTSI index [predicate] [value] % to [value] points.

In examples (2) and (3) we presented the templates for independent clauses, but in (4) there are two clauses connected by the linking word. The words in the square brackets represent the missing constituents or the slots of the templates.

The next step is filling in the slots in the templates with the words from our lexicon. The information about the types of index changes, or the events, affected the contents and the structure of the lexicon (both predicates and connective words) that is used by the program. There are groups of lexemes which characterize the changes and correspond to particular events. For example there are such groups as 'negative change predicates', e.g., *to fall*, *to decrease*, 'positive change predicates', e.g., *to rise*, *to increase*, 'no change predicates', e.g., *to remain constant*, etc. There are also such groups of words like 'nouns of change' related to the verbs, e.g., *rise*, *growth*, and 'intensifiers', e.g., *considerably*, *a lot*, etc.

Since we generate the news about two indexes which are indepent and can have different changes on the same day, it appears to be highly important to use plenty of connectives to provide the fluency to the texts. When the sentence templates had been already chosen and the most of the slots were filled in, the program applies the rules of templates combining. For example, if in the template (4a) the predicate is ‘to increase (by)’ and in (4b) it is ‘to fall (by)’, then the program chooses the adversative conjunction as a link for these clauses. In gerenal the choice of the connectives depends on the correlation of changes that two indexes demonstrate. It is taken into account if the indexes have the same change tendencies or they differ in their behavior and how much differ in it.

When all the previous steps are finished the program does the post-processing such as adding the punctuation marks and capitalisation where it is needed.

4 Evaluation

The system evaluation was divided into two stages, because the modules were evaluated separately.

The analysis module evaluation was done in the following way. For 100 data samples of index changes during the day we automatically determined their events and then manualy checked how many of these were determined correctly. The percentage of the right answers we got was 87%.

The NLG module was evaluated both manually and automatically using the BLEU metric (Papineni et al., 2002). For the manual evaluation we took 100 generated texts. These texts were rated according to the following scale: 2 — ‘fluent’, 1 — ‘understandable’, 0 — ‘disfluent’. It turned out that 61% of the texts were fluent, 28% were understandable and 5% were disfluent. The BLEU value appeared to be 0.66, for the calculation we used 70 gold standard sentences and 50 automatically generated sentences that describe the index changes. We decided to pick them for evaluation because unlike the other sentences in the news the sentences about changes have a high level of variation. We also admit the lack of gold standard material might have affected the BLEU results.

5 Acknowledgements

I would like to thank Alexei Nesterenko, PhD, for his advice, help and encouragement while I was doing the math for this research, I am also very thankful to Anastasia Bonch-Osmolovskaya, PhD, for her support and help during the whole time I worked on this paper and to Andrei Babitsky for his expert opinion on what the output news texts should be like.

References

Blake Howald, Ravi Kondadadi, and Frank Schilder. 2013. Domain adaptable semantic clustering in statistical nlg. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)*, pages 143–154.

Karen Kukich. 1983. Design of a knowledge-based report generator. In *Proceedings of the 21st annual meeting on Association for Computational Linguistics*, pages 145–150. Association for Computational Linguistics.

James H Martin and Daniel Jurafsky. 2000. Speech and language processing. *International Edition*, 710.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Ehud Reiter, Robert Dale, and Zhiwei Feng. 2000. *Building natural language generation systems*, volume 33. MIT Press.

Ilya Sutskever, James Martens, and Geoffrey E Hinton. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1017–1024.

Kees Van Deemter, Emiel Krahmer, and Mariët Theune. 2005. Real versus template-based natural language generation: A false opposition? *Computational Linguistics*, 31(1):15–24.