Noetic End-to-End Response Selection with Supervised Neural Network Based Classifiers and Unsupervised Similarity Models

Paweł Skórzewski, Weronika Sieińska, Marek Kubis
Adam Mickiewicz University in Poznań
Faculty of Mathematics and Computer Science
ul. Umultowska 87, 61-614 Poznań, Poland
pawel.skorzewski@amu.edu.pl, wersie@st.amu.edu.pl, mkubis@amu.edu.pl

Abstract

The paper describes a solution for the Noetic End-to-End Response Selection challenge – one of the tasks of the 7th Dialog System Technology Challenge. The goal of the task is to select the most appropriate continuation of a dialog from a given set of responses. We approach this problem by building the ensemble of supervised neural network based classifiers and unsupervised similarity models. The dialog continuation is selected according to the score that aggregates rankings of candidate responses determined by models that participate in the ensemble.

Introduction

In the recent years dialog systems have been gaining more and more popularity in both industry and research. Increasing number of approaches to the dialog management created the need for applicable comparison methods. As a response to this issue several initiatives have been started such as the BABI Dialogue Tasks (Li et al. 2016; Bordes, Boureau, and Weston 2017), a Spoken Dialogue Challenge 2010 (Black et al. 2010) and the Dialog System Technology Challenge (formerly the Dialog State Tracking Challenge) which is a series of tasks for estimating a user's goal in a spoken dialog system.

The traditional approach to creating goal-oriented dialog systems is to build a pipeline of separate modules for natural language understanding, dialog state tracking, action selection and natural language generation. The opposite approach – building end-to-end dialog systems – is increasingly gaining popularity and has become the subject of the 7th Dialog Systems Technology Challenge (DSTC7). In the end-to-end approach, the conversation model is trained on dialogs directly.

One of the most important goals of a dialog system that can be formulated as an end-to-end task is to provide the user with a relevant and comprehensive answer to the user's question. The response of the system should also sound as natural as possible. This goal can be achieved by choosing the best response from a given set of prepared sentences. The task formulated in this way is the subject of the first track of DSTC7.

This paper presents a system for Noetic End-to-End Response Selection challenge and reports its performance according to the DSTC7 Track 1 evaluation criteria (Yoshino et al. 2018).

The DSTC7 Track 1 was called Noetic End-to-End Response Selection Challenge. Its goal was to create an end-to-end system that for a given conversation and a given set of candidate responses selects a sentence that would be the best continuation of the dialog. The task was divided into several subtasks differing in terms of i.a. the number of all candidates and the number of correct ones. The overall aim of the challenge was to search for goal-oriented dialog systems solutions focused on the following aspects:

- language diversity (selecting a response from a rich set of human-generated paraphrases),
- large number of choices (candidate responses),
- varying number of expected correct responses (sometimes more than one of the candidate responses were correct, sometimes none of them),
- knowledge grounding (in some subtasks participants were given an additional dataset representing external knowledge),
- automation of the process (the solutions shouldn't use hand-crafted rules).

There were also two datasets given by the organizers: Ubuntu Dataset and Advising Dataset. The Ubuntu Dialog Corpus (Kummerfeld et al. 2018) is a new version of the large corpus of disentangled dialogs from IRC channel of Ubuntu Linux distribution's technical support, 25 times larger than that from Lowe et al. (2015). The Advising Corpus is a set of dialogs between students and their advisors talking about courses students should take in the curriculum.

The submissions were supposed to determine a ranking of utterances being candidates for responses for the provided dialogs. The ranking was expected to contain 100 candidates for the next utterance along with the corresponding confidence values. Furthermore, the candidates were required to be sorted in the order of confidence so that the most promising ones are placed first.

System overview

Our solution consists of an ensemble of several supervised and unsupervised models that rank the candidate responses independently followed by a voting procedure that determines the final result. We have combined two neural network based classifiers:

- Dual LSTM Encoder described as a baseline for the DSTC7 Sentence Selection Track,
- Dual GRU Encoder built as a modification of the aforementioned LSTM Dual Encoder, by substituting LSTM cells with GRU cells.

Furthermore, we have incorporated into the ensemble three classes of unsupervised similarity models developed independently for every dialog:

- TF-IDF – term frequency–inverted document frequency models (Spärck Jones 1972),
- LSI – latent semantic indexing models (Deerwester et al. 1990),
- PV – paragraph vector models (Le and Mikolov 2014).[1]

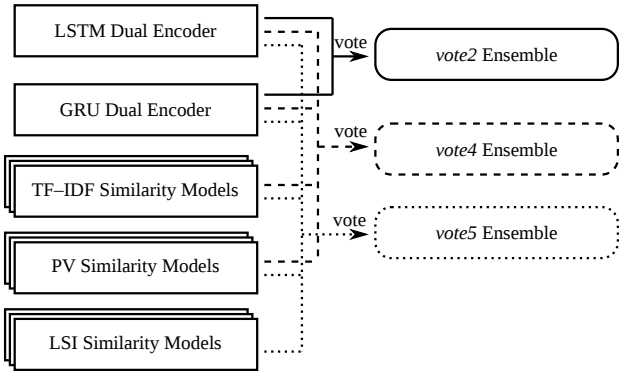

Figure 1: System Components

The data for the LSTM and GRU based classifiers was preprocessed using the script provided by organizers that included a simple tokenizer. For the unsupervised similarity models we have used preprocessing pipeline that consisted of a tokenizer, a lemmatizer and a delexicalizer. All processors of the pipeline work on `utterance` elements of the source input file. The first processor is a tokenizer, which is based on the Penn Treebank Tokenizer (Marcus, Marcinkiewicz, and Santorini 1993). The second processor is a lemmatizer, which is a WordNetLemmatizer from NLTK toolkit (Bird, Klein, and Loper 2009). The final stage of the preprocessing is delexicalization, which consists in replacing particular tokens of specific kind, e.g. URLs or e-mail addresses, with relevant placeholders (e.g. `#URL`, `#EMAIL`). The purpose of delexicalization is to focus on such types of tokens rather than particular values during learning the model.

The LSTM Dual Encoder model is used as a baseline for the DSTC7 Sentence Selection Track. It has been reported (Kadlec, Schmid, and Kleindienst 2015) that this model gives quite good performance on the original version of the Ubuntu dataset (Lowe et al. 2015). The architecture of LSTM Dual Encoder is based on Recurrent Neural Network model described by Lowe et al. (2015) and uses Long-Short Term Memory units (Hochreiter and Schmidhuber 1997): First, both the responses and the context are embedded into vectors using word embeddings. Then both vectors are put into the same LSTM neural network that generates a vector representation of the context and of the response. Matrix multiplication of the context representation and the matrix of trainable weights is then used to predict a response. The measure of similarity between the predicted response and the expected response is then used to update weights of the parameter matrix.

The GRU Dual Encoder model is quite identical to the LSTM Dual Encoder, with the exception that Gated Recurrent Units (GRU) are used rather than LSTM units. GRU (Cho et al. 2014) can be considered a variation on the LSTM that can produce comparable results for sequence-based tasks with long-term dependencies (Chung et al. 2014).

Both Dual Encoder models have been implemented using TensorFlow machine learning framework (Abadi et al. 2015). Due to hardware performance limitations, the number of epochs, embedding dimension and batch size had to be limited.

The unsupervised similarity models are trained on the per sample basis. For every dialog we prepare a separate corpus that consists of utterances being candidates for next turn responses. Then, TF-IDF, LSI and PV models are trained on the basis of this potential response corpus. Finally, we concatenate the utterances that form the dialog history and measure the similarity of the resulting word vector against the trained models. Thus, one can interpret the candidate responses as documents and the dialog history as a query being executed against the document collection in accordance to the information retrieval terminology. The result of matching the dialog history against the potential responses consists of a set of similarity scores that are used to form the candidate ranking.

We decided to incorporate the TF-IDF model into the ensemble since it encodes the information about the importance of words that are document-specific. In the TF-IDF model documents and queries become vectors of word-based features. A document (or query) vector consists of a sequence of weights determined for all words that apear in the corpus. The weights are computed according to the following formula:

$$f_w \times log_2 \frac{N}{d_w} \tag{1}$$

where f_w is the frequency of the word w in the given document (term frequency) and $\frac{N}{d_w}$ is the number of documents in the corpus divided by the number of documents that contain the word w (inverted document frequency).

[1]For training TF-IDF, LSI and PV models we use Gensim (Řehůřek and Sojka 2010). The PV models are called Doc2vec in Gensim.

The LSI model is obtained from the TF-IDF model by applying singular value decomposition (SVD) to the word occurrence matrix that consists of document vectors. The performed procedure reduces dimensionality of document vectors in order to utilise latent relationships among words and documents (Deerwester et al. 1990) for the purpose of document retrieval. In the experiments reported on the following pages we have used SVD to reduce the number of dimensions to 500.

Since TF-IDF and LSI procedures do not take into consideration the surrounding terms while determining weights for the individual words, we decided to extend the ensemble with paragraph vectors proposed in (Le and Mikolov 2014) which encode contextual information. In the PV model vectors for individual words are trained to maximize the following criterion (Le and Mikolov 2014)

$$\frac{1}{T}\sum_{t=k}^{T-k}\log p(w_t|w_{t-k},...,w_{t+k}) \tag{2}$$

where w_t stands for the t-th word in the sequence and k determines the size of the context being considered.[2] The word vectors are combined with the vectors representing individual documents[3] to become features for the softmax classifier. Le and Mikolov (2014) describe also the Bag-of-Words variant of their model that neglects the word order, but we did not adopt it since our main motivation to use paragraph vectors was to introduce sequential information into the ensemble.

The ensemble ranks candidate responses by aggregating results from the individual models. For every dialog we obtain five rankings – two from LSTM and GRU networks trained on the entire dataset and three from TF-IDF, LSI and PV models trained separately for every dialog. We assign scores to candidate responses on the basis of these model-specific rankings. The score of the candidate response c is calculated by the formula

$$\sum_{m\in M} r_c^m \tag{3}$$

where M is the set of models that participate in the ensemble and r_c^m is the position of the response c in the ranking inferred by the model m.[4] The responses are ordered according to this score to determine their positions in the final ranking returned by the ensemble.

Experiments and evaluation

For evaluation purposes, we prepared different versions of ensembles:

- ensemble of 2 models (LSTM and GRU – denoted as *vote2* in the tables),
- ensemble of 4 models (LSTM, GRU, TF–IDF and PV – denoted as *vote4* in the tables),
- ensemble of 5 models (LSTM, GRU, TF–IDF, LSI and PV – denoted as *vote5* in the tables).

We evaluated also the performance of all of the models solo (non-ensembled).

We submitted three solutions to the organizers: solo TF–IDF model, ensemble "vote4" and ensemble "vote5". The official evaluation results published by the organizers are denoted as *submitted* in the tables with results. These are best results according to the organizers chosen from the three solutions submitted by us, without distinction between individual systems ("tf–idf" vs. "vote4" vs. "vote5").

After we submitted the results to the organizers, we found a bug in a script for calculating the unsupervised similarity models results. That is why the results from *submitted* system differ from the results for individual models given in the tables.

The results of the experiments are shown in the Tables 1–8 on pages 4–5. Each table presents the results for a given subtask (1–5) on a given dataset (Ubuntu, Advising-Case-1 or Advising-Case-2). The "method" column indicates which model has been used to obtain the results:

- The "solo" models are denoted by one of the labels: *lstm*, *gru*, *tfidf*, *lsi*, *pv*.
- Ensemble models are denoted by one of the labels: *vote2*, *vote4*, *vote5*.
- The label *submitted* denotes the result for model submitted to the organizers for evaluation.
- The results denoted as *best* are the best results for each subtask and dataset published by the organizers of the challenge. The organizers published only two metrics for the best systems: Recall@10 and MRR, and they did not reveal what those best systems were. Therefore some values in the tables are missing.

The best results for each of the metrics (Recall@1, Recall@10, Recall@50, MRR, and MAP for subtask 3), excluding model denoted as *best*, have been highlighted in bold in the tables. As can be seen in the tables, an ensemble of two models: LSTM Dual Encoder and GRU Dual Encoder (*vote2*), gave the best results for subtask 1 on Ubuntu dataset. For other subtask-and-dataset combination, TF–IDF model performed the best, although all these results are barely comparable with the best results published by the organizers, the main reason being probably the aforementioned hardware performance problems that we have encountered.

[2] We used contexts of 150 words ($k = 75$) for the purpose of conducting experiments.

[3] Called paragraphs in (Le and Mikolov 2014).

[4] For instance, the score of a candidate response that appears at the second place in the GRU ranking and at the first place in all the other rankings is 6.

References

Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Goodfellow, I.; Harp, A.; Irving, G.; Isard, M.; Jia, Y.; Jozefowicz, R.; Kaiser, L.; Kudlur, M.; Levenberg, J.; Mané, D.; Monga, R.; Moore, S.; Murray, D.; Olah, C.; Schuster, M.; Shlens, J.; Steiner, B.; Sutskever, I.; Talwar, K.; Tucker, P.; Vanhoucke, V.; Vasudevan, V.; Viégas,

method	Recall@1	Recall@10	Recall@50	Recall@100	MRR	MAP
lstm	0.006	0.093	**0.518**	1.000	0.046	0.046
gru	0.009	0.089	0.506	1.000	0.049	0.049
tfidf	0.008	0.015	0.027	0.029	0.011	0.011
lsi	0.008	0.008	0.008	0.008	0.008	0.008
pv	0.002	0.008	0.019	0.029	0.004	0.004
vote2	**0.011**	**0.097**	0.496	1.000	**0.053**	0.053
vote4	0.000	0.008	0.024	0.029	0.003	0.003
submitted	0.008	0.008	0.008	N/A	0.008	N/A
best	N/A	0.902	N/A	N/A	0.735	N/A

Table 1: Results for subtask 1 on dataset Ubuntu

method	Recall@1	Recall@10	Recall@50	Recall@100	MRR	MAP
lstm	0.006	0.080	0.404	0.798	0.039	0.039
gru	0.007	0.092	0.394	0.798	0.042	0.042
tfidf	**0.072**	**0.151**	0.242	0.280	**0.101**	0.101
lsi	**0.072**	0.072	0.072	0.072	0.072	0.072
pv	0.015	0.062	0.183	0.280	0.034	0.034
vote2	0.005	0.071	**0.411**	0.798	0.037	0.037
vote4	0.014	0.081	0.222	0.280	0.036	0.036
submitted	**0.072**	0.072	0.072	N/A	0.072	N/A
best	N/A	0.739	N/A	N/A	0.589	N/A

Table 2: Results for subtask 4 on dataset Ubuntu

method	Recall@1	Recall@10	Recall@50	Recall@100	MRR	MAP
lstm	0.006	0.104	0.498	1.000	0.050	0.050
gru	0.006	0.092	0.490	1.000	0.046	0.046
tfidf	**0.064**	**0.248**	**0.652**	1.000	**0.131**	0.131
lsi	**0.064**	0.064	0.064	0.064	0.064	0.064
pv	0.022	0.142	0.556	1.000	0.072	0.072
vote2	0.014	0.108	0.510	1.000	0.056	0.056
vote4	0.022	0.174	0.604	1.000	0.081	0.081
vote5	0.022	0.174	0.604	1.000	0.081	0.081
submitted	**0.064**	0.064	0.064	N/A	0.064	N/A
best	N/A	0.850	N/A	N/A	0.608	N/A

Table 3: Results for subtask 1 on dataset Advising-Case-1

method	Recall@1	Recall@10	Recall@50	Recall@100	MRR	MAP
lstm	0.036	0.318	**0.848**	1.000	0.126	0.122
gru	0.036	0.334	0.840	1.000	0.131	0.126
tfidf	**0.084**	**0.354**	0.814	1.000	**0.178**	**0.206**
lsi	**0.084**	0.084	0.084	0.084	0.084	0.057
pv	0.034	0.212	0.638	1.000	0.098	0.118
vote2	0.048	0.334	**0.848**	1.000	0.138	0.129
vote5	0.054	0.294	0.826	1.000	0.137	0.143
submitted	0.048	0.334	**0.848**	N/A	0.138	0.129
best	N/A	0.906	N/A	N/A	0.624	N/A

Table 4: Results for subtask 3 on dataset Advising-Case-1

method	Recall@1	Recall@10	Recall@50	Recall@100	MRR	MAP
lstm	0.008	0.074	0.350	0.766	0.038	0.038
gru	0.012	0.066	0.352	0.766	0.041	0.041
tfidf	**0.048**	**0.188**	**0.502**	0.766	**0.098**	0.098
lsi	**0.048**	0.048	0.048	0.048	0.048	0.048
pv	0.028	0.118	0.446	0.766	0.064	0.064
vote2	0.006	0.062	0.350	0.766	0.035	0.035
vote4	0.020	0.108	0.468	0.766	0.058	0.058
vote5	0.022	0.108	0.468	0.766	0.059	0.059
submitted	0.006	0.062	0.352	N/A	0.035	N/A
best	N/A	0.652	N/A	N/A	0.350	N/A

Table 5: Results for subtask 4 on dataset Advising-Case-1

method	Recall@1	Recall@10	Recall@50	Recall@100	MRR	MAP
pv	0.012	0.148	0.566	1.000	0.063	0.063
tfidf	**0.034**	**0.228**	**0.628**	1.000	**0.103**	0.103
best	N/A	0.630	N/A	N/A	0.339	N/A

Table 6: Results for subtask 1 on dataset Advising-Case-2

method	Recall@1	Recall@10	Recall@50	Recall@100	MRR	MAP
pv	0.032	0.210	0.614	1.000	0.094	0.112
tfidf	**0.042**	**0.340**	**0.808**	1.000	**0.139**	**0.160**
best	N/A	0.750	N/A	N/A	0.434	N/A

Table 7: Results for subtask 3 on dataset Advising-Case-2

method	Recall@1	Recall@10	Recall@50	Recall@100	MRR	MAP
pv	0.006	0.098	0.448	0.816	0.046	0.046
tfidf	**0.026**	**0.172**	**0.492**	0.816	**0.079**	0.079
best	N/A	0.508	N/A	N/A	0.242	N/A

Table 8: Results for subtask 4 on dataset Advising-Case-2

F.; Vinyals, O.; Warden, P.; Wattenberg, M.; Wicke, M.; Yu, Y.; and Zheng, X. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Bird, S.; Klein, E.; and Loper, E. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Black, A. W.; Burger, S.; Conkie, A.; Hastie, H.; Keizer, S.; Lemon, O.; Merigaud, N.; Parent, G.; Schubiner, G.; Thomson, B.; Williams, J. D.; Yu, K.; Young, S.; and Eskenazi, M. 2010. Spoken dialog challenge 2010: Comparison of live and control test results.

Bordes, A.; Boureau, Y.-L.; and Weston, J. 2017. Learning end-to-end goal-oriented dialog. *International Conference on Learning Representations*.

Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.

Deerwester, S.; Dumais, S. T.; Furnas, G. W.; Landauer, T. K.; and Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41(6):391–407.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Kadlec, R.; Schmid, M.; and Kleindienst, J. 2015. Improved deep learning baselines for ubuntu corpus dialogs. *CoRR* abs/1510.03753.

Kummerfeld, J. K.; Gouravajhala, S. R.; Peper, J.; Athreya, V.; Gunasekara, C.; Ganhotra, J.; Patel, S. S.; Polymenakos, L.; and Lasecki, W. S. 2018. Analyzing assumptions in conversation disentanglement research through the lens of a new dataset and model. *ArXiv e-prints*.

Le, Q. V., and Mikolov, T. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, 1188–1196.

Li, J.; Miller, A. H.; Chopra, S.; Ranzato, M.; and Weston, J. 2016. Dialogue learning with human-in-the-loop. *arXiv preprint arXiv:1611.09823*.

Lowe, R.; Pow, N.; Serban, I.; and Pineau, J. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the SIGDIAL 2015 Conference, The 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2-4 September 2015, Prague, Czech Republic*, 285–294.

Marcus, M. P.; Marcinkiewicz, M. A.; and Santorini, B. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics* 19(2):313–330.

Řehůřek, R., and Sojka, P. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 45–50. Valletta, Malta: ELRA. `http://is.muni.cz/publication/884893/en`.

Spärck Jones, K. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* 28(1):11–21.

Yoshino, K.; Hori, C.; Perez, J.; D'Haro, L. F.; Polymenakos, L.; Gunasekara, C.; Lasecki, W. S.; Kummerfeld, J.; Galley, M.; Brockett, C.; Gao, J.; Dolan, B.; Gao, S.; Marks, T. K.; Parikh, D.; and Batra, D. 2018. The 7th Dialog System Technology Challenge. *arXiv preprint*.