Skip-Thought Vector based Chatbots

Paul Mierau
Chair for Data Processing, Technical University of Munich
paul.mierau@tum.de

***Abstract*—This paper explores the potential of using Skip Though sentence encoding as part of an intent based chatbot system. Skip-Thought vectors constitute a state of the art approach of sentence embedding that aims to allow machines to extract the meaning of a sentence.**
For that purpose a support vector machine classifier is trained on a dataset of 64 intents using Skip-Thought encoding as a feature generator. The implementation is tested against industry leading platforms such as IBM Watson, Google Dialogflow or Cognigy AI.
Despite being built with a much lower complexity and using significantly less resources, the Skip-Thought vector based chatbot ranks 3 out of 5 with respect to accuracy and F1-Score, showing the high potential Skip-Thought embedding provides.

Keywords*—*Skip-Thought Vectors, Chatbots, Natural Language Understanding, Intent Classifiers.

I. Introduction

The vast majority of human knowledge is stored and accessible in the form of written language. Teaching machines how to efficiently access and use this type of information has been the objective of natural-language understanding (NLU), a subcategory of natural-language processing (NLP), for several decades.
The results, that are at the cutting edge of these efforts are continuously becoming part of our daily lives. Conversational agents (chatbots), that automate customer relations in areas like retail or customer support constitute a quiet established application based on NLU research. Despite the successful implementations of its current development stage, the language understanding capabilities of these systems are still very limited [1].

The approach of Skip-Thought Vectors, promises to be a groundbreaking attempt to model natural reasoning based on written language [2]. In the Skip-Thoughts model, an encoder-decoder architecture is used for unsupervised learning of a generic sentence encoder. For the learning, a special loss function is implemented, which extends the skip-gram model of so called word vectors [3]. Instead of mapping the individual words in a sentence to a vector representation, an entire sentence gets encoded. By predicting surrounding sentences from continuous text, the loss implementation aims at modeling context and hence a sort of reasoning into the encoding. While Skip-Thought Vectors have been tested in their capability to measure semantic relatedness, perform paraphrase detection and basic classification [2], their suitability for building production capable chatbots was unexplored.

This paper considers the question of how a chatbot, based on Skip-Thought sentence embedding, performs compared to current leading chatbot vendors. For that purpose a skip-thought vector encoder is implemented and embedded into a support vector machine based classifier. The classifier is then trained and tested using a dataset and benchmarking method from the paper Benchmarking Natural Language Understanding Services for building Conversational Agents [4]. In a final step the results are compared to a selection of leading conversational agent service platforms such as Google Dialogflow, IBM Watson and Cognigy.

II. State of the Art

The declared objective of a chatbot is to allow user interactions with a backend system, or an automated service, using-natural language. The most important component of a modern, in production chatbot systems is the natural-language understanding engine (NLU engine). The core of an NLU engine is its intent classifier, which maps an incoming user query to a so called intent [5]. An intent constitutes a predefined category of user requests, that specifies the goal of the interaction. Once a user intent is classified, the system is able to evaluate the next steps based on the user query. Eventually a suitable response is chosen from either a knowledge base or generated from the result of an API request, as can be seen in figure 1

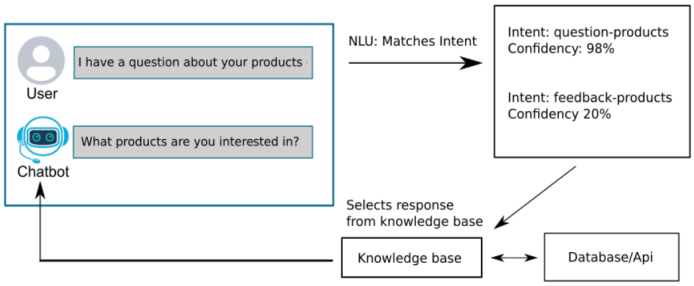

Fig. 1: An overview of the user-chatbot interaction process

The intent classifier is the result of either some sort of unsupervised or supervised learning on a dataset [6]. In the more common supervised example, a dataset consists of example sentences for a user query and their corresponding intent label.

The process of intent matching involves a feature generator and a classifier [7] as illustrated by figure 2 .

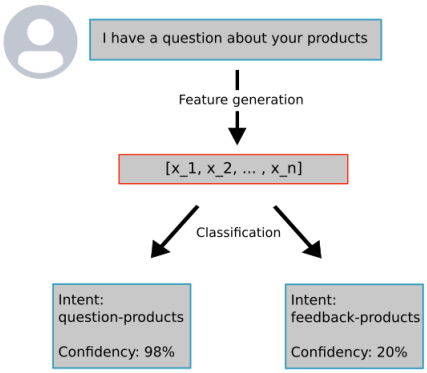

Fig. 2: Process of intent matching

During the last decade, feature generators for intent classification have developed from n-gram based architectures over to word vector (word2vec) algorithms, setups using convolutional neural networks (CNNs) or transformer models such as BERT [8]. The goal of the feature extraction is to convert a sentence, which consists of a sequence of letters, into a sequence of numbers (vector) that represents meaningful attributes such as semantic information, of that sentence within the scope of a language.
As for classifier however, a variety of models and approaches have been established suitable for applications in the field of NLU. Amongst the most popular setups are maximum entropy methods, K-nearest neighbors algorithms, support vector machines (SVMs) and long short-term memory networks (LSTMs) [9].

III. Problem Statement

Intent classification extends the problem of data classification by a problematic nature of its target data. Language itself tends to be an inefficient way of exchanging information, especially when applied in a context of social activities.[10]
Ambiguity in interpretations, relevance of context, complexity and semantic layers of languages are amongst the factors that make intent classification a challenging task. On top of that, out of vocabulary and unseen intent handling needs to be taken into consideration.[8]
In general however, the ability to grasp the meaning of a query and thus to consider the context of words in a sentence, has shown to be the most important factor of intent classification. [7] This is emphasized through the ground breaking improvements that came along with BERT.[11]
Further improving feature generation thus remains one of the key tasks to enhance chatbot capabilities.

IV. Approach

The chatbot build in this paper uses Skip-Thought encoding for feature generation from queries and a SVM model for intent classification. A full NLU interface and pipeline is implemented, which allows for intent definition and training as well as prompting a query.

A. Skip-Thought Encoder

A Skip-Thought encoder is generated using an encoder-decoder architecture. Given a large dataset of contiguous text, following the idea of unsupervised learning, the encoder attempts to convert a sentence s_i to a numerical representation, while two decoders proceed in recreating the previous sentence s_{i-1} as well as the following sentence s_{i+1}. The choice for an encoder-decoder architecture was made on recurrent neural networks (RNNs) using a gated recurrent unit (GRU), as this has been shown to perform well [2].

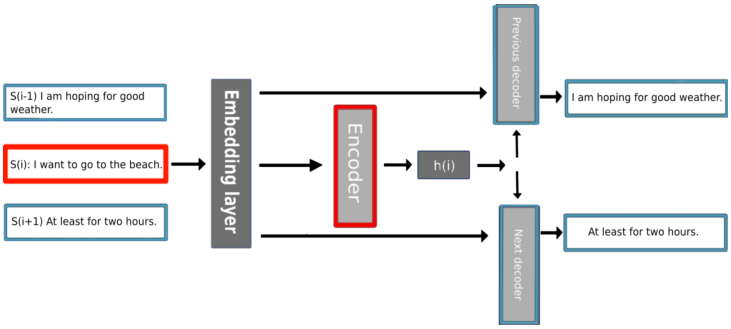

Fig. 3: The Skip-Thought model scheme

As can be seen in figure 3 the hidden state h_i of the encoding layer is used as output of the encoder.

Let x^t denote a vector $\mathbb{R}^{M\times 1}$ at time $t = 1, ..., N$, containing the words $w_i^1, ..., w_i^N$ from sentences s_i $i = 1, ..., M$ of length N and W_r, W_z, U_r, U_z be corresponding weight matrices, then at each timestep t, the encoder computes the following sequence of equations in order to receive hidden state h^t.

$$r^t = \sigma(W_r x^t + U_r h^{t-1}) \quad (1)$$
$$z^t = \sigma(W_z x^t + U_z h^{t-1}) \quad (2)$$
$$\bar{h}^t = \Phi_h(W x^t + U(r^t \odot h^{t-1}) \quad (3)$$
$$h^t = (1 - z^t) \odot h^{t-1} + z^t \odot \bar{h}^t \quad (4)$$

The initial hidden state h_0 is set to 0. The hidden state h_i^N is then taken as a the feature vector of s_i.
This paper implements two such encoders. One which processed a sentence from beginning to the end (UniSkip) and another which looks at the sentence in reversed order as well (BiSkip).
For the UniSkip a hidden Unit size of 2400 is chosen. The

BiSkip encoder is build with a hidden Unit size of 1200 for each forward and backward processing. Finally a vector of dimension 4800 is formed out of the output from the two encoders.

The decoders used for training the encoders follow the same logic, except for an additional weight matrix that induces the input of the encoder into the set of above equations. Given the feature vector h_i of sentence s_i the goal is to optimize the log-probabilities for sentences s_{i-1} and s_{i-1}:

$$\sum_{t}^{K}(logP(w_{i+1}^{t}|w_{i+1}^{<t},h_i)) + \sum_{t}^{L}(logP(w_{i-1}^{t}|w_{i-1}^{<t},h_i)) \quad (5)$$

where K and D denote the length of the surrounding sentences. The probability of a sentence is given by:

$$\sum_{t}^{K}(logP(w_{i+1}^{t}|w_{i+1}^{<t},h_i)) \propto exp(v_{w_{i+1}^{t}}h_{i+1}^{t}) \quad (6)$$

where h_{i+1}^{t} is the output of a decoder and V the vocabulary matrix [2].
Since it turned out that training a sentence feasible Skip-Thought encoder would extend the ressources of this paper, a pre-trained model from the University of Toronto was used. The model was trained on the bookCorpus [12], containing 11,038 books from 16 different genres and is the most properly trained model publicly available up to this day.

B. Support Vector Machine classifier

The choice for the intent classifier was made on a support vector machine model for mainly two reasons.
SVMs haven been proven to be quite effective in the field of intent classification throughout several experiments [8] [9]. Thus the quality of the classification would be high enough to not compromise the comparison of performance against state-of-the art chatbot platforms. Its simple architecture and setup however ensure the focus being on the feature extraction capabilities of the Skip-Thought encoder and allow for easy reproducibility of the experiment.
In particular the python package *sklearn* is used to implement its default *Nu-SVC* classifier with a linear kernel.

V. EXPERIMENT

The evaluation of the generated Thought-Vector chatbot is orientated on the paper Benchmarking Natural Language Understanding Services for building Conversational Agents [4], as it has been used by several vendors to benchmark their own product. On top of that the project provides a publicly available dataset, designed specifically to test in production chatbot NLU engines.

A. Dataset

The dataset consists of a set for training the NLU as well as a seperate set for testing. Both contain a collection of 64 intents based on a home-assistant bot scenario. While the original training dataset includes 190 example queries per intent class, this paper only uses a total of 30 sentences for each class, to train the classifier. This choice was made in order to make the test results more accounting to real, in production situations and aligns with similar attempts to compare market leaders. [13] Table 1 shows example queries from the training set.

Intent class	Query
general_explain	could you again elaborate me on your answers please
general_explain	could not understand it
calendar_remove	delete all my appointments for today
calendar_remove	cancel sam's party
iot_hue_lightdim	i want the lights less brighter
iot_hue_lightdim	bed room two darken

TABLE I: Example queries from the training dataset

The dataset for testing includes a total of 5518 queries approximately 86 sentences per intent class.
By using less queries per class for training than for testing, an additional layer of approximation towards a real life scenario is achieved.

B. Metrics

Two measurements are taken in order to evaluate the results. Each of which provides a slightly different inside on the performance of the chatbot.
The first measurement is the *accuracy* of the classifier, which describes the ratio between the number of all correctly classified intents and all tested queries. Accuracy measurements are widely used in the field of NLU and thus allow for a direct comparison with most vendors.
The second measurement is the so called *macro F1 score*.
The F1 score is derived by first tracking the True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN) per intent class.
Using these results a recall and precision value can be calculated by:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

For each class the corresponding F1 score is now obtained by:

$$F1 = 2 * \frac{(Precision * Recall)}{(Precision + Recall}$$

Taking measurements separately per class, using a One-vs-Rest (OvR) strategy is necessary in multi-class classification. Following the macro average approach the overall F1 score is received by the arithmetic mean of all separate scores.
The motivation to include F1 scores for the classifier evaluation, is based on its capability to account for imbalances with respect to FP and FN values. For chatbots a wrongly classified positive is more expensive than a false negative. Hence a high precision value is required. The recall value on the other hand allows for a more precise estimate of the NLUs competence of grasping the intent behind a query than the accuracy.

C. Comparisons

The performance measurements of the implemented Skip-Thoughts chatbot are compared against the results of testing four different chatbots, based on the NLUs of leading vendors in the market.
This is done to receive a direct placement in the scope of the current market, such that the feasibility of Skip-Thought encoders for chatbot instances can be realistically assest. All bots got trained and tested using the same datasets.
Corresponding measurements for the Googles Dialogflow, Microsoft Luis and IBMs Watson have been obtained by Benchmarking Natural Language Understanding Services [4]. Results for Cognigy AI are provided by the company itself [13].

D. Results

	Skip-Thoughts Chatbot	Cognigy AI	Dialogflow	LUIS	Watson
Accuracy	0.778	0.846	0.761	0.788	0.81
F1(macro)	0.779	0.827	0.758	0.776	0.804

TABLE II: Results of the experiment

Table 2 shows the performance results of the experiment.
With respect to accuracy as well as the F1 Score the following placement is obtained:

1) Cognigy AI
2) Watson
3) Skip-Thoughts Chatbot
4) LUIS
5) Dialogflow

It is notable however, that only the Skip-Thoughts Chatbot does not have a lower F1 Score than the corresponding accuracy. By directly comparing these two measurements such a result is only possible if the classifier has identified a higher amount of TP than TN. Given that a dataset in intent classification consists of more negative examples, a high FP rate for one ore two classes can be concluded.

VI. Conclusion

When comparing performance of the Skip-Thoughts Chatbot in relation to market leading vendors, two factors need to be taken into account. The complexity of this papers classifier is rather simple compared to it's professional competition and the resources available for its training were vastly disproportionate.
In this context a Rank of 3 out of 5 appears to be a remarkable result.
On closer inspection the observed high FP rate is not desirable. As stated in this paper a wrongly classified positive is more expensive in an average chatbot customer relation. Since the F1 Score is a macro average over all classes it is to be assumed that one or two classes come with such a high sensitivity.
In order to improve these results a more sophisticated classifier would be the next step. Additionally it is worth exploring the effects of including the intent examples in the Skip-Thought encoder generation already.

As a final conclusion Skip Thought vectors have once more proven their potential in the field of natural language understanding.
A potential next step would be to explore the suitability of Skip-Thought vectors for self learning chatbots.

REFERENCES

[1] E. Almansor and F. Hussain, *Survey on Intelligent Chatbots: State-of-the-Art and Future Research Directions*, 01 2020, pp. 534–543.

[2] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler, "Skip-thought vectors," *CoRR*, vol. abs/1506.06726, 2015. [Online]. Available: http://arxiv.org/abs/1506.06726

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *Proceedings of Workshop at ICLR*, vol. 2013, 01 2013.

[4] X. Liu, A. Eshghi, P. Swietojanski, and V. Rieser, "Benchmarking natural language understanding services for building conversational agents," 03 2019.

[5] A. Abdellatif, K. M. S. Badran, D. E. Costa, and E. Shihab, "A comparison of natural language understanding platforms for chatbots in software engineering," *ArXiv*, vol. abs/2012.02640, 2021.

[6] A. Chatterjee and S. Sengupta, "Intent mining from past conversations for conversational agent," *CoRR*, vol. abs/2005.11014, 2020. [Online]. Available: https://arxiv.org/abs/2005.11014

[7] P. Alonso, "Faster and more resource-efficient intent classification," Ph.D. dissertation, Lule University of Technology, 2020. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:ltu:diva-81178

[8] H. Weld, X. Huang, S. Long, J. Poon, and S. C. Han, "A survey of joint intent detection and slot-filling models in natural language understanding," *CoRR*, vol. abs/2101.08091, 2021. [Online]. Available: https://arxiv.org/abs/2101.08091

[9] R. Sarikaya, G. E. Hinton, and A. Deoras, "Application of deep belief networks for natural language understanding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 778–784, 2014.

[10] H. Purohit, G. Dong, V. Shalin, K. Thirunarayan, and A. Sheth, "Intent classification of short-text on social media," 12 2015, pp. 222–228.

[11] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[12] Y. Zhu, R. Kiros, R. S. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," *CoRR*, vol. abs/1506.06724, 2015. [Online]. Available: http://arxiv.org/abs/1506.06724

[13] D. Seisser. (2020) Benchmarking nlu engines: A comparison of market leaders. Accessed: 2022-15-01. [Online]. Available: https://www.cognigy.com/blog/benchmarking-nlu-engines-comparing-market-leaders

[14] J. Zhang, K. Hashimoto, Y. Wan, Y. Liu, C. Xiong, and P. S. Yu, "Are pretrained transformers robust in intent classification? A missing ingredient in evaluation of out-of-scope intent detection," *CoRR*, vol. abs/2106.04564, 2021. [Online]. Available: https://arxiv.org/abs/2106.04564