Skip Act Vectors: integrating dialogue context into sentence embeddings

Jeremy Auguste, Frédéric Béchet, Geraldine Damnati, Delphine Charlet

Skip Act Vectors: integrating dialogue context into sentence embeddings

Jeremy Auguste, Frédéric Béchet, Géraldine Damnati and Delphine Charlet

Abstract This paper compares several approaches for computing dialogue turn embeddings and evaluate their representation capacities in two dialogue act related tasks within a hierarchical Recurrent Neural Network architecture. These turn embeddings can be produced explicitely or implicitely by extracting the hidden layer of a model trained for a given task. We introduce *skip-act*, a new dialogue turn embeddings approach, which are extracted as the common representation layer from a multi-task model that predicts both the previous and the next dialogue act. The models used to learn turn embeddings are trained on a large dialogue corpus with light supervision, while the models used to predict dialog acts using turn embeddings are trained on a sub-corpus with gold dialogue act annotations. We compare their performances for predicting the current dialogue act as well as their ability to predict the next dialogue act, which is a more challenging task that can have several applicative impacts. With a better context representation, the *skip-act* turn embeddings are shown to outperform previous approaches both in terms of overall F-measure and in terms of macro-F1, showing regular improvements on the various dialogue acts.

1 Introduction

Following the successful application of continuous representation of words into vector spaces, or *embeddings*, in a large number of Natural Language Processing tasks [14][15], many studies have proposed the same approach for larger units than words such as sentences, paragraphs or even documents [10][11]. In all cases the

Jeremy Auguste and Frédéric Bechet
Aix-Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France, e-mail: firstname.lastname@lis-lab.fr

Géraldine Damnati and Delphine Charlet
Orange Labs, Lannion, France, e-mail: firstname.lastname@orange.com

main idea is to capture the *context of occurrence* of a given unit as well as the unit itself.

When processing dialog transcriptions, being able to model the *context of occurrence* of a given turn is of great practical use in applications such as automated dialog system for predicting the next action to perform, or analytics in order, for example, to pair questions and answers in a corpus of dialog logs. Therefore finding the best embedding representations for dialog turns in order to model dialog structure as well as the turns themselves is an active field of research.

In this paper, we evaluate different kinds of sentence-like (turns) embeddings on dialogue act classification tasks in order to measure how well they can capture dialog structures. In a first step, the dialogue turn embeddings are learned on large corpus of chat conversations, using a light supervision approach where dialogue act annotations are given by an automatic DA parser. Even if the annotations are noisy, this light supervision approach allows us to learn turn-level vector representations on a large amount of interactions. In a second step, the obtained turn-level vector representations are used to train dialogue act prediction models with a controlled supervised configuration.

After presenting the dialogue act parser architecture in Section 3, we will present the various dialogue turn embeddings approaches in Section 4. The corpus and the dialogue act annotation framework are presented in Section 5 while Section 6 describes the experimental results.

2 Related Work

In order to create and then evaluate the quality of embeddings, several different types of approaches have been proposed. For word embeddings, a lot of work has been done to try to evaluate how they are able to capture relatedness and similarity between two words by using manual annotation [9][12][4] and by using cognitive processes [18][2]. However, on sentence embeddings, it is not easy to tell how similar or related two sentences are. Indeed, the context in which they appear is very important to truly understand the meaning of a sentence and how it interacts with other sentences.

Multiple papers propose different kinds of evaluation tasks in order to evaluate different kinds of sentence embeddings. In [8], the authors use the SICK [13] and STS 2014 [1] datasets to evaluate the similarity between sentences by using similarity ratings. They also use sentiment, opinion polarity and question type tasks to evaluate the embeddings. As these datasets are composed of sentence pairs without context, the proposed sentence embeddings approaches are only based on the sentence itself. In [7], sentence embeddings are evaluated by looking at their ability to capture surface, syntactic and semantic information. Here again, this framework primarily focuses on the sentence itself and not on the context in which it is produced. In [5], a sentence embeddings evaluation framework is proposed that groups together most of the previous evaluation tasks in addition to inference, captioning and

paraphrase detection tasks. In all of the above approaches, the focus is on the evaluation of sentence embeddings such as Skip-thoughts [10], ParagraphVectors [11] or InferSent [6] in order to find out the embeddings that have the best properties in general. However, none of these embeddings and evaluation tasks are built to take into account dialogues and more specifically, the structure and interactions in a dialogue. Some work has been done in order to take into account the dialogue context in [17]. In their work, the authors try to take into account this context by using a modified version of word2vec to learn sentence embeddings on dialogues. These embeddings are then evaluated by comparing clusters of sentence embeddings with manually assigned dialogue acts. This allows to see if the learned embeddings capture information about the dialogue context, however it does not use explicit dialogue structure information to learn the embeddings. In our work, we use a corpus with a noisy dialogue act annotation to learn specialized sentence embeddings that try to directly capture information about the context and interactions in the dialogue.

3 Dialogue Act Parser Architecture

In order to be able to create sentence embeddings that take into account the dialogue context, we will be using dialogue acts. They allow us to partially represent the structure and the interactions in a dialogue. We use two different kinds of models to parse these dialogue acts where one kind is used to create sentence embeddings, while the second kind is used to later evaluate the different embeddings.

The first architecture is a 2-level hierarchical LSTM network where the first level is used to represent the turns in a conversation, and the second level represents the conversation, as shown in Figure 1. The input is the sequence of turns which are themselves sequences of words represented as word embeddings. The word embeddings are trained by the network from scratch. The dialogue acts are predicted using the output for each turn at the second level. Since we do not use a bidirectional LSTM, the model only makes use of the associated turn and the previous turns of a conversation in order to predict a given act. It has no information about the future, nor about the previous acts. This architecture allows us to use the hidden outputs of the first layer as the sentence embeddings of each turn.

The second architecture is a simple LSTM network which only has a single layer, as shown in Figure 2. The input sequence that is given to the LSTM is the sequence of turns of a conversation where each turn is replaced by a pre-trained turn embedding. For each turn, the corresponding output in the LSTM is used to predict its dialogue act. This architecture is the one used to evaluate the different kinds of fixed pre-trained embeddings that are described in Section 4.

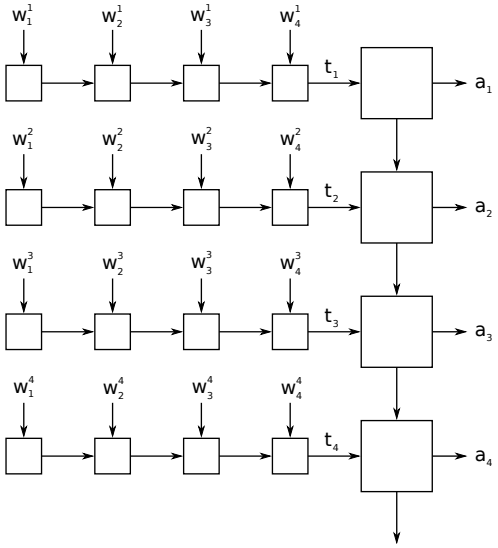

Fig. 1: Two level LSTM architecture used to create embeddings. w_i^j is the word i of turn j, t_j is the learned turn embedding and a_j is the predicted act.

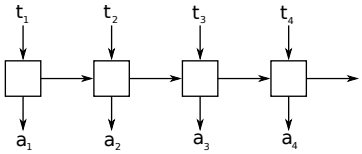

Fig. 2: LSTM architecture used for evaluation. t_i is a fixed pre-trained turn embedding and a_i is the predicted act.

4 Skip-Act vectors

It is possible to construct sentence embeddings using several different means, each of them being able to capture different aspects of a sentence. In our case, we want to find out what kind of embeddings are the best at capturing information about the dialogical structure and the context in which appears a turn. Multiple different kind of embeddings are thus trained on the `DATCHA_RAW` corpus (the large unannotated corpus described in section 5. The following self-supervised embeddings are trained:

Word Average This is simply the average of all the word embeddings in the turn. The word embeddings are learned with FastText [3] on the `DATCHA_RAW` corpus using a dimension of 2048 and a window size of 6. These can be considered as our baseline embeddings since they do not directly take into account the context in which the turns are produced.

Skip-thought These embeddings are learned using a skip-thought model [10]. This model tries to learn the sentence embeddings by trying to regenerate the adjacent sentences during the training. Thus, it tries to learn the context in which a sentence is produced.

In addition to these self-supervised embeddings, we also learned embeddings based on supervised tasks. To learn these embeddings, we use the 2-level LSTM architecture described in Section 3. The following supervised embeddings are trained:

RNN Curr Act These embeddings are learned by using a hierarchical neural network that is trained to predict the dialogue act of each turn. The embeddings are the hidden output from the turn layer of the network. Since the DATCHA_RAW corpus is not annotated with dialogue acts, we used a system developed during the DATCHA[1] project based on a CRF model developed in [16] (85.7% accuracy) to predict the dialogue acts of each turn of the corpus.

RNN Next Act These embeddings are created similarly to the RNN Curr Act embeddings but instead of predicting the current act for a given turn, the following act is instead predicted.

RNN Prev Act These embeddings are created similarly to the RNN Curr Act embeddings but instead of predicting the current act for a given turn, the previous act is instead predicted.

Skip-Act These embeddings combine the ideas of RNN Prev Act and RNN Next Act by using the same turn layer in the network for both tasks. This model shares the idea of the Skip-thought vectors by trying to learn the context in which the turns are produced. But instead of trying to regenerate the words in the adjacent turns, we try to predict the dialogue acts of the adjacent turns. This allows us to hope that the learned embeddings will focus on the dialogue context of turns. The architecture of this model is presented in Figure 3.

5 Corpus

Chat conversations are extracted from Orange's customer services contact center logs, and are gathered within the DATCHA corpus, with various levels of manual annotations. The DATCHA corpus covers a wide variety of topics, ranging from technical issues (e.g. solving a connection problem) to commercial inquiries (e.g. purchasing a new offer). They can cover several applicative domains (mobile, internet, tv).

For our experiments, we use two different subsets of these chats:

- Chats from a full month that do not have any gold annotation (79000 dialogues, 3400000 turns) (DATCHA_RAW);
- Chats annotated with gold dialogue act annotation (3000 dialogues, 94000 turns) (DATCHA_DA)

[1] http://datcha.lif.univ-mrs.fr

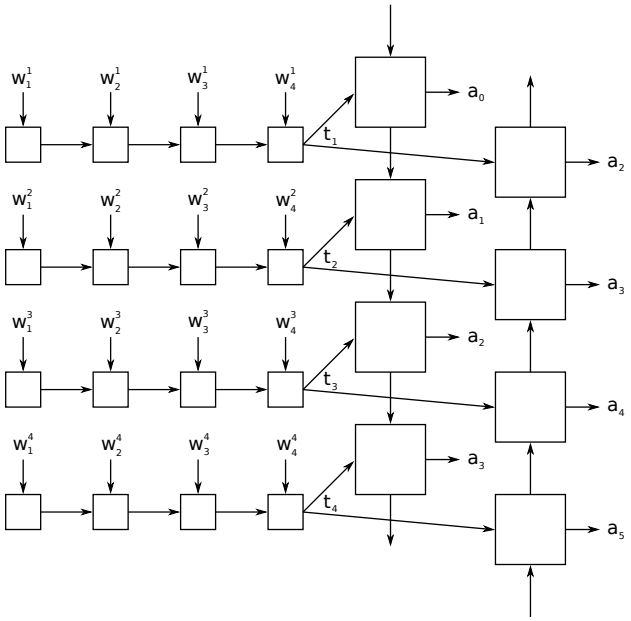

Fig. 3: Architecture used to create skip-act vectors. w_i^j is the word i of turn j, t_j is the learned turn embedding and a_j is the predicted act.

These subsets are partitioned into train, test and development parts. The label set used in the dialogue act annotation is as follows:

Label	Meaning	Description
OPE	Opening	Opening turns in the dialogue
PRO	Problem description	The client's description of his problem
INQ	Information question	Turn where a speaker asks for some information
CLQ	Clarification question	A speaker asks for clarification
STA	Statement	New information input
TMP	Temporisation	Starting a break of the dialogue
PPR	Plan proposal	Resolution proposal of the problem
ACK	Acknowledgement	A speaker acknowledges the other speaker's sayings
CLO	Closing	Closing turn
OTH	Other	For turns that don't match other described labels

This set has been designed to be as generic as possible, while taking into account some particular aspects of professional chat interactions (e.g. *Problem description* or *Plan proposal*). The distribution of the different types of dialogue acts in the test split of the DATCHA_DA corpus can be found in Figure 4. We also indicate the distributions when considering only a single speaker since they use very different types of turns. For instance, *Plan proposals* are almost exclusively uttered by Agents while, conversely, *Problem descriptions* are mostly observed on Customers side.

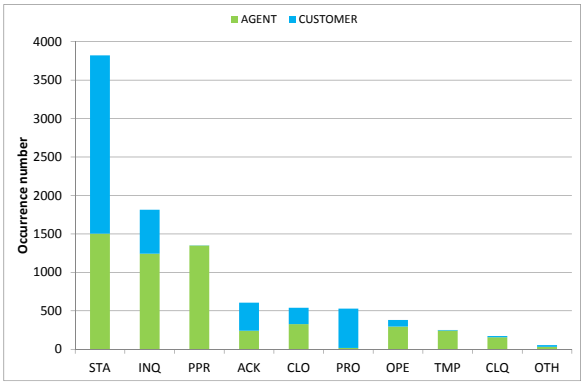

Fig. 4: Dialogue act distribution in the DATCHA_DA test corpus

6 Turn embeddings evaluation

6.1 Evaluation protocol

We want to make sure that the generated embeddings are able to capture the different aspects of a dialogue. Dialogue acts are one way to partially represent the structure and interactions in a dialogue. Thus, we evaluate the different embeddings on two tasks. For the first task, we try to predict the dialogue act of a turn by only using the sequence of embeddings of the current and previous turns. For the second task, we do the same thing but instead of predicting the dialogue act of the current turn, we predict the act of the next turn (without giving the embedding of the next turn in the input). This second task allows us to tell if the learned embeddings manage to capture information about not only the turn but also about the context in which these turns are produced.

Some of the created embeddings are learned using tasks that involve dialogue acts, thus it is likely that these embeddings obtain the best results. But it is interesting to see if other embeddings are able to obtain similar or close results.

For both tasks, we use the architectures described in Section 3 with a hidden size of 512. For each turn, the corresponding output in the RNN is given to a decision layer which uses a softmax to output a probability distribution of the dialogue acts. We use cross-entropy as our loss function and Adam as the optimizer with a learning rate of 0.001. The PyTorch framework is used to build the different architectures.

In order to evaluate the quality of the different predictions, we primarily use 2 metrics:

- **accuracy**: the percentage of correct decisions;

- **macro F1**: the non-weighted average of the F1-measures of the 10 act labels. The F1-measure is the harmonic mean of precision P and recall R for a given label l such as $F1(l) = \frac{2 \times P(l) \times R(l)}{P(l)+R(l)}$;

6.2 Results and Analyses

LSTM architecture	pre-trained embeddings	Current Act		Next Act	
		Accuracy	Macro-F1	Accuracy	Macro-F1
2-level hierarchical	None	83.69	78.15	46.21	26.45
turn level	Word Average	82.96	79.47	48.26	30.09
turn level	Skip-thought	82.50	75.73	48.30	28.61
turn level	RNN Curr Act	84.74	80.47	48.54	31.42
turn level	RNN Next Act	84.40	81.42	49.97	34.47
turn level	RNN Prev Act	83.02	80.44	48.77	31.96
turn level	**Skip-act**	**85.24**	**82.16**	**49.96**	**35.33**

Table 1: Evaluation of the prediction of the current and next dialogue acts on all turns

Results of the prediction of the current and next acts are reported in Table 1. The first line corresponds to the first model described in Figure 1 where no pre-trained embeddings are used and where the embeddings are learned jointly with the model's parameters on the DATCHA_DA corpus. The following lines correspond to the single turn-level architecture presented in Figure 2 using several variants of fixed turn embeddings, pre-trained on the large DATCHA_RAW corpus. For each embedding type and task, we only report the results of the configuration that obtained the best results. We can first note a big difference in performances between the two tasks with the next act task being much harder than the current act task. It seems to be very difficult to predict the next act given the history of turns, particularly for some of them, as can be seen in Figure 5 and Figure 6 where some acts such as CLQ, INQ or PPR see a drop of 60 points in their F1-score while acts such as STA, CLO or OPE only have a drop of 20 points. This could be explained by the fact that closings and openings are easier to locate in the conversation, while statements are the most represented labels in conversations. On the other hand, it is not necessarily easy to know that the next turn is going to be a question or a plan proposal. We can also notice that the OTH act is not at all correctly predicted in the next act task, and even in the current act task it is the label with the worst F1-score. This is probably due to the fact that turns that are labeled OTH are usually filled with random symbols or words and are both very diverse and not frequent.

Unsurprisingly, for both tasks, the best results are obtained with embeddings learned using dialogue acts. However, the **Word Average** and **Skip-thought** vec-

tors both achieve good results but they still are 2 points lower than the best results. It is interesting to note that the **Skip-thought** vectors do not achieve better results than **Word Average** vectors on the next act task. This can be surprising since they would have been expected to better capture information about the surrounding turns, however the generalization from word level prediction to turn level prediction is not sufficiently efficient. It is also interesting to note that better results are achieved by **RNN Curr Act** embeddings (84.74%), which are learned on a corpus with a noisy annotation, compared to results achieved by the embeddings learned during the training on the DATCHA_DA corpus (83.69%) which has gold annotation. This results confirms our choice to train turn embeddings separately with light supervision on a significantly larger, even though noisy, training corpus.

Another interesting aspect of these results is the comparison of the different kinds of embeddings learned with dialogue act related tasks. Indeed, on the current act task, we can notice that **RNN Curr Act** embeddings obtain slightly lower results (-0.5 points) than **Skip-act** embeddings. This is surprising since **RNN Curr Act** are learned using the same task than the evaluation, while **Skip-act** are learned by trying to predict the next and previous acts only. These results could mean that **Skip-act** are more robust since they learn in what context the acts are produced. On the next act task, both the **RNN Next Act** and **Skip-act** achieve the same performances with 50% accuracy, while the **RNN Curr Act** embeddings obtain an accuracy of 48.5%.

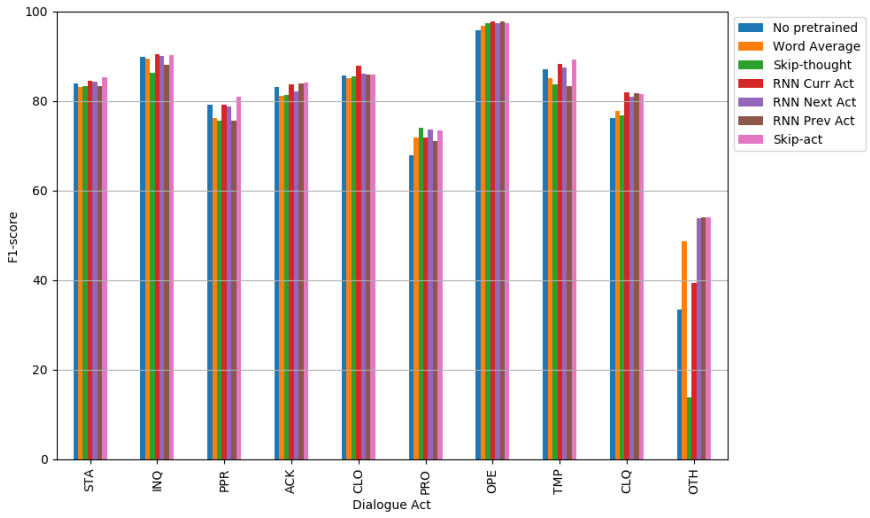

Fig. 5: F1-scores on the current act task on all turns

We also reported in Table 2 and 3 the results when considering only the turns from respectively the agent and the client for evaluation. It is important to note that the label distribution is very different depending on the speaker. Most of the

LSTM architecture	pre-trained embeddings	Current Act		Next Act	
		Accuracy	Macro-F1	Accuracy	Macro-F1
2-level hierarchical	None	84.22	77.38	35.87	23.16
turn level	Word Average	82.48	77.31	37.78	27.02
turn level	Skip-thought	80.36	74.75	37.07	25.39
turn level	RNN Curr Act	84.70	79.01	38.90	29.00
turn level	RNN Next Act	84.30	82.42	41.29	32.60
turn level	RNN Prev Act	83.24	80.11	38.80	28.81
turn level	**Skip-act**	**85.48**	**82.94**	**42.30**	**33.56**

Table 2: Evaluation of the prediction of the current and next dialogue acts on agent's turns

LSTM architecture	pre-trained embeddings	Current Act		Next Act	
		Accuracy	Macro-F1	Accuracy	Macro-F1
2-level hierarchical	None	83.01	58.58	59.48	21.13
turn level	Word Average	83.59	60.97	61.71	21.80
turn level	Skip-thought	85.31	59.13	62.70	20.49
turn level	RNN Curr Act	84.78	64.16	60.89	21.74
turn level	RNN Next Act	84.54	63.20	61.09	22.91
turn level	RNN Prev Act	82.74	61.88	61.56	21.73
turn level	**Skip-act**	**84.93**	**63.99**	**59.78**	**23.79**

Table 3: Evaluation of the prediction of the current and next dialogue acts on customer's turns

questions (`CLQ` and `INQ`) and nearly all plan proposals (`PPR`) and temporisations (`TMP`) are from the agent while most of the problem descriptions (`PRO`) and the majority of statements (`STA`) are from the client. When evaluated on the agent side, **Skip-act** embeddings are again the best embeddings for both tasks, being 1 point higher than the **RNN Next Act** embeddings and 3.5 points higher than the **RNN Curr Act** embeddings. These results are interesting since the agent is the speaker with the most variety in the types of turns, including many turns with questions, plan proposals or temporisations. This seems to indicate that **Skip-acts** manage to capture more information about the dialogue context than the other embeddings. We can also notice that this time, **Skip-thought** vectors obtain lower results than the simple **Word Average**. When evaluated on the customer side, **Skip-thought** vectors obtain the best scores on both tasks when looking at the accuracy (85.31% and 62.70%) but lower scores in terms of macro-F1. The scores on the next act task are higher but this is only due to the fact that the `STA` act represents 57.4% of the samples, whereas on all the turns and for the agent they respectively represent 40.2% and 27.8% of the samples.

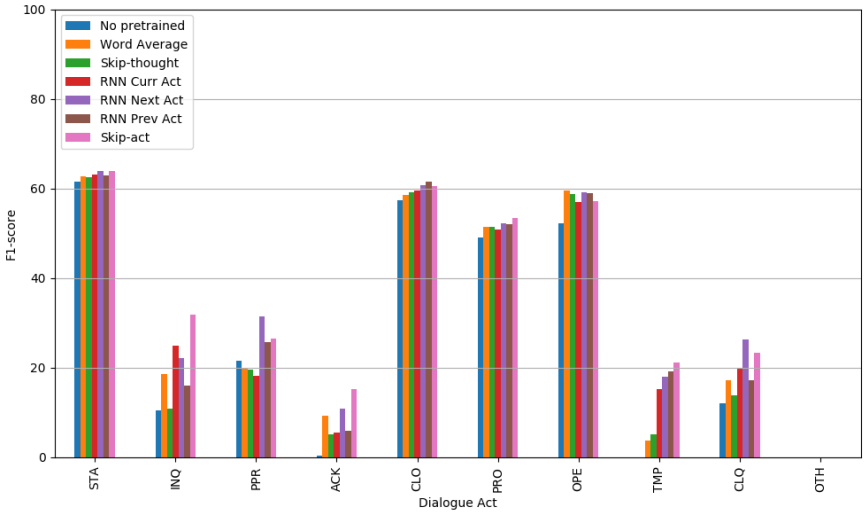

Fig. 6: F1-scores on the next act task on all turns

7 Conclusion

We have proposed a new architecture to compute dialogue turn embeddings. Within the skip-act framework, a multitask model is trained in order to jointly predict the previous and the next dialogue acts. Trained in a lightly supervised way on a large corpus of chat conversations with an automatic dialogue act annotation, the output of the common hidden layer provides an efficient turn level vector representation that tends to capture the dialogic structure of the interactions. We have evaluated several dialogue turn embeddings configurations on two tasks, first predicting the associated dialogue act of the current turn, and then predicting the next dialogue act which is a more challenging task requiring a better representation of the dialogue structure. Skip-act embeddings achieve the best results on both tasks. In the future, it would be interesting to combine skip-thoughts and skip-acts in order to be able to capture the semantic and syntactic information in addition to the dialogue context of turns.

Acknowledgements This work has been partially funded by the Agence Nationale pour la Recherche (ANR) through the following programs: ANR-15-CE23-0003 (DATCHA), ANR-16-CONV-0002 (ILCB) and ANR-11-IDEX-0001-02 (A*MIDEX).

References

1. Agirre, E., Banea, C., Cardie, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W., Mihalcea, R., Rigau, G., Wiebe, J.: Semeval-2014 task 10: Multilingual semantic textual similarity. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 81–91 (2014)
2. Auguste, J., Rey, A., Favre, B.: Evaluation of word embeddings against cognitive processes: Primed reaction times in lexical decision and naming tasks. In: Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP, pp. 21 – 26. Copenhagen, Denmark (2017)
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching Word Vectors with Subword Information. Transactions of the Association of Computational Linguistics **5**(1), 135–146 (2017)
4. Bruni, E., Boleda, G., Baroni, M., Tran, N.K.: Distributional semantics in technicolor. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, pp. 136–145. Association for Computational Linguistics (2012). 00120
5. Conneau, A., Kiela, D.: SentEval: An Evaluation Toolkit for Universal Sentence Representations. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). Miyazaki, Japan (2018)
6. Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A.: Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 670–680 (2017)
7. Conneau, A., Kruszewski, G., Lample, G., Barrault, L., Baroni, M.: What you can cram into a single \$&!#* vector: Probing sentence embeddings for linguistic properties. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 2126–2136. Association for Computational Linguistics, Melbourne, Australia (2018)
8. Hill, F., Cho, K., Korhonen, A.: Learning Distributed Representations of Sentences from Unlabelled Data. In: Proceedings of NAACL-HLT, pp. 1367–1377 (2016)
9. Hill, F., Reichart, R., Korhonen, A.: Simlex-999: Evaluating semantic models with (genuine) similarity estimation. Computational Linguistics (2016). 00173
10. Kiros, R., Zhu, Y., Salakhutdinov, R.R., Zemel, R., Urtasun, R., Torralba, A., Fidler, S.: Skip-thought vectors. In: Advances in neural information processing systems, pp. 3294–3302 (2015)
11. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning, pp. 1188–1196 (2014)
12. Luong, T., Socher, R., Manning, C.D.: Better word representations with recursive neural networks for morphology. In: CoNLL, pp. 104–113 (2013). 00192
13. Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., Zamparelli, R.: A SICK cure for the evaluation of compositional distributional semantic models. In: LREC, pp. 216–223 (2014)
14. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In Proceedings of Workshop at ICLR (2013). 03267
15. Pennington, J., Socher, R., Manning, C.D.: Glove: Global Vectors for Word Representation. In: EMNLP, vol. 14, pp. 1532–1543 (2014). 01307
16. Perrotin, R., Nasr, A., Auguste, J.: Dialog Acts Annotations for Online Chats. In: 25e Conférence Sur Le Traitement Automatique Des Langues Naturelles (TALN). Rennes, France (2018)
17. Pragst, L., Rach, N., Minker, W., Ultes, S.: On the Vector Representation of Utterances in Dialogue Context. In: LREC (2018)
18. Søgaard, A.: Evaluating word embeddings with fMRI and eye-tracking. ACL 2016 p. 116 (2016). 00000