# To BERT or Not to BERT Dealing with Possible BERT Failures in an Entailment Task

Pedro Fialho[1,3(✉)], Luísa Coheur[1,2], and Paulo Quaresma[1,3]

[1] INESC-ID Lisboa, Lisbon, Portugal
peter.fialho@gmail.com
[2] Instituto Superior Tecnico, Universidade de Lisboa, Lisbon, Portugal
[3] Universidade de Évora, Évora, Portugal

**Abstract.** In this paper we focus on an Natural Language Inference task. Being given two sentences, we classify their relation as NEUTRAL, ENTAILMENT or CONTRADICTION. Considering the achievements of BERT (Bidirectional Encoder Representations from Transformers) in many Natural Language Processing tasks, we use BERT features to create our base model for this task. However, several questions arise: can other features improve the performance obtained with BERT? If we are able to predict the situations in which BERT will fail, can we improve the performance by providing alternative models for these situations? We test several strategies and models, as alternatives to the standalone BERT model in the possible failure situations, and we take advantage of semantic features extracted from Discourse Representation Structures.

**Keywords:** Natural Language Inference · Feature engineering · Failure prediction model

## 1 Introduction

Natural Language Inference (NLI) is a known task in Natural Language Processing (NLP)[1]. It can be implemented as a classification task in which the model needs to decide about the relation between a pair of sentences. Usual categories are ENTAILMENT, NEUTRAL and CONTRADICTION.

BERT (Bidirectional Encoder Representations from Transformers) [7] is a state-of-the-art language model that has shown impressive performance on many NLP tasks. Here, we take advantage of BERT to perform NLI. However, we also implement other NLI classifiers, based on lexical and semantic features that we extract from the Discourse Representation Structures obtained for each pair of sentences we want to classify. Then, we implement two strategies to detect possible failures. The first is based on the fact that BERT has lower results in ENTAILMENT and CONTRADICTION situations. Therefore, we run BERT and directly accept the NEUTRAL labels, while other classifiers are employed


M.-J. Lesot et al. (Eds.): IPMU 2020, CCIS 1237, pp. 734–747, 2020.
https://doi.org/10.1007/978-3-030-50146-4_54

in the other cases. In addition, we also implement several models that try to predict when BERT will fail. In the latter cases, other models are employed. Results show that we can improve results with the models based on lexical and semantic features.

This paper is organized as follows: Sect. 2 presents related work and Sect. 3 our models. Section 4 describes the experimental setup and Sect. 5 the results. Finally, Sect. 6 presents the main conclusions and future work.

## 2 Related Work

A benchmark for systems aimed at Recognizing Textual Entailment (RTE) was initially developed in the PASCAL challenge series [2]. The RTE task is to detect entailment between a premise and an hypothesis, while a related task is to detect NLI, where target labels are ENTAILMENT, CONTRADICTION and NEUTRAL (no semantic relation).

NLI is represented in the SICK corpus [13], composed by 10000 pairs of sentences, seeded from corpora of image and video captions, and expanded by rule based transformations to introduce particular linguistic phenomena, such as negations. SICK is annotated by crowd-sourcing, and was the target of a shared task on the Semeval evaluation series [12].

Following SICK, the much larger SNLI [5] corpus was released, containing 570000 examples also seeded from a corpus of captions and annotated by crowd-sourcing, but instead expanded by crowd-sourcing. SNLI inspired the creation of other corpora on NLI, for instance the e-SNLI corpus [6] that augments SNLI with natural language explanations for the annotations, or the MultiNLI corpus [21], that follows the same design procedure and size of SNLI, but instead of captions includes sentence pairs from other text genres and sources, such as fiction books or transcripts of conversations. MultiNLI is one of the targets of the GLUE benchmark [20], that evaluates systems for their joint performance on multiple Natural Language Understanding (NLU) tasks.

Various forms of assessing NLI are presented in the mentioned shared tasks and benchmarks. However, as modern machine learning architectures particularly leverage large data collections, recent approaches suitable for NLI are mostly applied to corpora such as SNLI or MultiNLI, both for their greater size and complexity. One of such approaches is the BERT model [7].

BERT generates a dynamic embedding according to the context in which a word is employed, and may even generate the embedding of a sentence pair, if the aim is to verify entailment on the pair [7]. Training a BERT model is expensive on time and resources, but models based on Wikipedia were made available in its original release.

The BERT model achieves competitive results on various NLU tasks, as shown from its performance on the GLUE benchmark [7], but also specifically in NLI, such as when applied only to MultiNLI [7], to SNLI [22], or to the recent CommitmentBank corpus [10] which is part of the SuperGLUE benchmark [19], that supersedes GLUE.

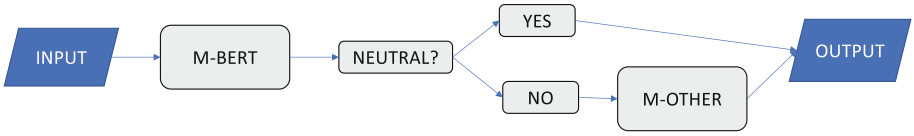


**Fig. 1.** Simple model – M-BERT directly used to detect NEUTRAL relations.

Recent studies on the generalization of various models, including BERT, suggest that performance is only consistent when assessed within the same benchmark [18], from combining train and test sets of different corpora. Other works focus specifically on BERT failures in NLI, such as in [14] to hypothesize that the success of BERT relies on the occurrence of certain linguistic patterns in the data, or in [10] to suggest that BERT does not implicitly learn linguistic priors and is mostly driven by statistical regularities. To the best of our knowledge, the performance of BERT in the SICK corpus was not yet evaluated.

## 3 Entailment and Failure Models

In this section we describe the models we use to perform the NLI task and the strategies we have implemented to predict when BERT will fail.

The BERT model, trained to perform NLI, uses BERT embeddings as features. From now on we will call M-BERT to this model. A set of lexical and semantic features, alone or associated with BERT embeddings, are also used to train several classifiers that perform NLI. We call M-OTHER to these models. Our semantic features are based on Discourse Representation Structures (DRS), that is, a formal representation of meaning that follows the Discourse Representation Theory [11].

Our first strategy (from now on Strategy 1) takes advantage of M-BERT results to decide which are the possible failure conditions. We have observed that BERT has lower results in both ENTAILMENT or CONTRADICTION situations. Thus, we run M-BERT and accept all the NEUTRAL labels, according to it. For the remaining labels we run the M-OTHER models, trained in the NLI task, but in a corpus that only has ENTAILMENT or CONTRADICTION labels. Figure 1 depicts this strategy.

We also implement a second strategy (from now on Strategy 2) in which we train several models that try to predict when BERT will fail. The previous mentioned lexical and semantic features, along with BERT, are used by these models. We call M-FAIL to these models. Here, the idea is the following: if a model of type M-FAIL predicts that BERT will fail, then the previous models, trained in the NLI task, are used instead of M-BERT. Figure 2 illustrates this strategy.

Finally, instead of using a single M-FAIL model to predict M-BERT failure, we consider the predictions of the different M-FAIL models. Three options are considered:

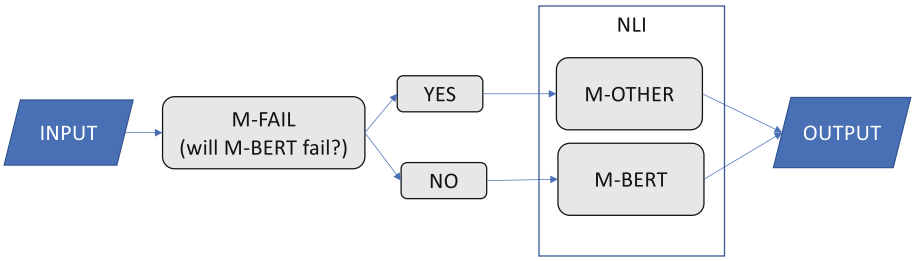


**Fig. 2.** Pipeline with M-FAIL models.

- *at least one*: if one model from the M-FAIL family returns an M-BERT failure, we will consider that M-BERT will fail;
- *majority voting*: if the majority of the models from the M-FAIL family returns an M-BERT failure, we will consider that M-BERT will fail;
- *all*: if all the models from the M-FAIL family returns a M-BERT failure, we will consider that M-BERT will fail

Considering the previous scheme, the different M-OTHER models will be used if M-BERT is expected to fail.

## 4 Experimental Setup

### 4.1 Corpora

Our experiments rely on the SICK corpus [13] for English. As previously said, sentences in SICK are image captions obtained by crowd-sourcing. Each instance in SICK, that is, each pair of sentences, is labelled as NEUTRAL, ENTAILMENT or CONTRADICTION regarding the semantic relation between the two sentences. For instance, the pair composed by the sentences "Three kids are jumping in the leaves" and "Three boys are jumping in the leaves", is labeled as ENTAILMENT, while the former sentence paired with "Three kids are sitting in the leaves" is labeled as NEUTRAL. An example of a pair labeled as CONTRADICTION in SICK is the pair composed by the sentences "Nobody is riding the bicycle on one wheel" and "A person is riding the bicycle on one wheel".

We follow the partitions suggested in [13], but 5 SICK instances were discarded as the DRS parser, Boxer [4], was unable to process them. Therefore, our train, development and test set have 4436, 495 and 4904 pairs of sentences, respectively. Notice that the train set is unbalanced, as 2522 pairs are labelled as NEUTRAL, 1274 as ENTAILMENT and 640 as CONTRADICTION.

**Balancing the Training Data.** In preliminary experiments, we have observed that when a negation was involved in a sentence, the classifiers found more difficult to return the appropriate label . In addition, we consider that a strong

lexical overlap could be easy to identify (at least by the models using the lexical features), and thus, that more complicated situations occur in scenarios of low lexical overlap between sentences. Therefore, we tried to balance the train set, in what respects these two characteristics (negation and low lexical overlap). We decided, then, to split the original training into 2 partitions with 50% each, by considering:

- the presence of a negation in at least one of the sentences of a pair, as identified with DRS semantics, and
- low lexical overlap, as identified by a Jaccard score lower than 0.6 or a BLEU score lower than 0.5.

In the 2522 NEUTRAL instances in the original train set, 416 have a negation and 2188 have low lexical overlap. In the 1274 ENTAILMENT instances, 10 have a negation and 634 have low lexical overlap. Finally, from the 640 CONTRADICTION instances, 575 have a negation and 318 have low lexical overlap. Hence, training instances that contain a negation are almost equally distributed among NEUTRAL and CONTRADICTION classes, and most of the examples from these classes have low lexical overlap. Negations are almost not employed in examples of the ENTAILMENT class, and there are as much examples with low lexical overlap as those with high lexical overlap. We split the original train set in two, each containing 50% of the examples from each class, and 50% of the examples that comply with the above features. For instance, the first set contains 319 examples of the CONTRADICTION class, of which 287 employ a negation and 166 have low lexical overlap.

**Building Corpora for Strategy 1 and 2.** In order to implement Strategy 1, the one that takes advantage of M-BERT results, we removed from the train corpus the NEUTRAL relation and train the M-OTHER models in order to distinguish ENTAILMENT from CONTRADICTION situations.

Concerning Strategy 2, and in order to create a reference to train the M-FAIL models (the FAIL-CORPUS), we split the training set in two (as previously described). In the first half we trained M-BERT. Then, we run it on the second half, to build the corpus to train the M-FAIL model: every time M-BERT successfully labelled an NLI relation, the associated sentence pair was labeled as 1; it was labeled as 0 otherwise. As usually, the development set was used for tuning (and first tests) and the test set for the final evaluation. Figure 3 details these partitions.

As we will see, since M-BERT model is successful in most examples, the dataset to train M-FAIL models is unbalanced. Therefore, to train M-FAIL models we discard examples where BERT succeeded until reaching the same number of examples where BERT failed to identify the entailment class, hence obtaining a balanced FAIL-CORPUS.

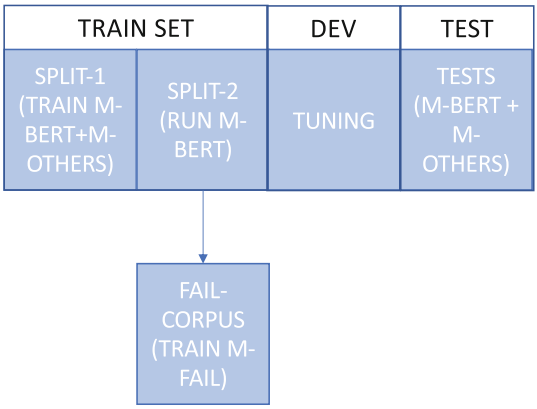


**Fig. 3.** Corpus partition.

### 4.2 Evaluation Metrics

The performance of our system on entailment detection is measured with Accuracy, Precision, Recall and F-measure (F1, as we consider precision and recall to have the same weight/importance). All metrics produce values between 0 and 1, where greater values are better, hence we report results in percentages.

As the entailment task on SICK configures a multi class classification setup, and the Precision, Recall and F1 metrics are based on the assumption that a positive label exists (as in binary classification), we calculate such metrics using an average of scores from binary classifications, one for each class such that the positive label represents belonging to the class. We chose to average by a weighted mean that considers the number of instances of each class, since class distribution is imbalanced in SICK.

Our definition of accuracy also considers class imbalance. In a multi class setting, the accuracy is defined per class, and obtained by dividing each element in the diagonal of the confusion matrix (true positives per class) by the sum of elements in the corresponding row (the total number of examples of a class). The balanced accuracy is the arithmetic mean of the per class accuracy values.

### 4.3 Features

**Lexical Features.** We employ the INESC-ID@ASSIN [9] system that generates almost 100 features for a pair of sentences, based on the lexical aspects of their words or by using some similarity measure. Examples of such features are the length of the longest sentence, or the BLEU [16] metric.

**Semantic Features.** We obtain DRSs from the Boxer framework [4], containing semantic aspects for each sentence, such as the implicit entities resulting from

pronoun resolution, or the type of a quantity, for instance to distinguish parts of a date from other numbers in a sentence.

Given two DRS, we compute 16 features that represent aspects shared by both or occurring in any of the DRS. These include: a) boolean features, such as to indicate the presence of a negation in any of the DRS; b) count based features, such as for the number of equivalent entities between the negated subsets of each DRS; c) percentage based features, such as the ratio of equivalent entities and total entities in both DRS, according to various entity comparison techniques, and; d) distance based features, such as from measuring the mean gap between dates from each DRS.

Entities within DRS are considered:

- not equivalent, if a word pair, one from each DRS, is an antonym in the WordNet [8] database;
- equivalent, if it is a synonym in WordNet;
- equivalent if the cosine of their FastText [3] embeddings is greater than 0.4. This threshold was chosen by observation, and as a compromise between the cosines for synonyms and antonyms sampled from WordNet.

Any technique for entity comparison results in 2 features, one for the count of entities matched and the other for the percentage of entities matched in the total count of entities of both DRS.

Other than entities, a DRS is also composed of conditions, defined as relations between a source and a target entity. We consider the target entities from a pair of conditions of the same type, one from each DRS, as equivalent if the source entities are also equivalent according to matched entities from the previously mentioned entity comparison techniques. Thus, relative to conditions, we consider two entities as equivalent if employed in the same type of condition, with the same role and paired with equivalent entities.

**BERT Embeddings.** We employ the base and uncased version of BERT pre-trained models for English only, as provided with the original BERT release[1], which produces embeddings with 768 dimensions. For such model, we lowercased text and removed accents from sentence pairs before input to BERT.

### 4.4 Tools and Model Configuration

Machine learning and data processing is mostly provided by scikit-learn [17]. All models are trained using Support Vector Machines (SVM) with a linear kernel, from the LIBLINEAR implementation. To obtain the final model for a certain combination of features, 7 different models are trained, corresponding to different values for the C parameter, sampled from a logarithmic scale between 0.001 and 1000. The model with optimal C parameter is further calibrated to maximize the performance of the SVM [15]. All model tuning is evaluated on the SICK development set.

[1] https://github.com/google-research/bert.

Lexical and semantic features are linearly scaled with various approaches, according to the type of feature or feature vector. For instance, for all feature vectors, values greater than 1 are scaled to the 0 to 1 range, while for feature vectors that include BERT we do not employ feature centering around zero, since BERT features are sparse.

## 5 Results and Discussion

As previously said, M-BERT and M-OTHER models were trained in the first partition of the training set and evaluated in the test set. M-FAIL models were trained in the FAIL-CORPUS. In this section, we will identify each model according to the features that they use; we will use "b" for BERT features, "l" for the lexical features and "d" for the DRS ones.

### 5.1 M-BERT and M-OTHER Results

Results obtained by M-BERT and M-OTHER models can be seen in Table 1.

**Table 1.** Performance in the entailment task of the different models.

| Features | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| b (M-BERT) | 78.62% | 80.47% | 80.53% | 80.46% |
| b+d | 79.57% | 81.16% | 81.18% | 81.13% |
| b+l | **79.98**% | 81.73% | 81.77% | **81.71**% |
| b+l+d | 78.56% | 79.96% | 79.87% | 79.89% |
| l | 67.78% | 74.96% | 75.18% | 74.58% |
| d | 74.16% | 75.99% | 76.06% | 75.93% |
| l+d | 76.72% | 78.92% | 78.92% | 78.79% |

The two best results differ from the others in at least 1% of accuracy, and almost the same for F1, and correspond to M-OTHER models trained on combinations of BERT embeddings with lexical or semantic features (b+l and b+d, respectively). M-BERT is the third best result.

Other than BERT features, the most informative features of the M-OTHER model based on semantic features include the previously described features for the count of matched entities according to DRS conditions and the percentage of matched entities from lexical semantics heuristics.

The most informative lexical features in the b+l model include various count based features, after scaled to the 0 to 1 range. The only non scaled feature in such set is the cosine distance between vector representations of trigram sequences for each sentence.

**Table 2.** STRATEGY 1.

| Features | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| b | 78.80% | 80.45% | 80.55% | 80.46% |
| b+l | 78.83% | 80.47% | 80.57% | 80.48% |
| b+d | 78.85% | 80.53% | 80.61% | 80.53% |
| b+l+d | 78.88% | 80.56% | 80.63% | 80.56% |
| l | 70.10% | 76.21% | 76.20% | 76.02% |
| d | **78.91%** | 80.85% | 80.79% | **80.76%** |
| l+d | **78.91%** | 80.85% | 80.79% | **80.76%** |

### 5.2 Strategy 1 Results

Table 2 shows the results obtained by following STRATEGY 1.

Of the 4904 instances in the test set, 58% were predicted as neutral by M-BERT, and the remaining were classified by models trained only on ENTAILMENT and CONTRADICTION instances.

The best result was obtained from the model based on semantic features, or lexical and semantic features combined, while the worst result, with less 4% of F1 performance, is from the model based only on lexical features. In the l+d model, the only semantic feature of its most informative set is the count of matched entities according to heuristics, while lexical features in this set are once again mostly count based features.

### 5.3 M-FAIL Results

Table 3 shows the results obtained by the different M-FAIL models.

**Table 3.** M-FAIL results

| Features | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| b | 58.20% | 84.78% | 60.86% | 70.86% |
| b+l | 59.12% | 84.94% | 65.61% | 74.03% |
| b+d | 59.21% | 85.00% | 65.48% | 73.97% |
| b+l+d | 59.28% | 85.00% | 65.91% | 74.25% |
| l | 59.48% | 84.87% | 69.11% | 76.19% |
| d | 58.47% | 84.11% | 73.41% | **78.39%** |
| l+d | **59.88%** | 85.01% | 70.03% | 76.80% |

M-BERT predicts the correct entailment class on 80% of the test set instances, hence the accuracy of M-FAIL models mostly represent their ability to predict that M-BERT will correctly identify the entailment class of a

given example, which is low. However, F1 is more robust to such imbalanced situations, since it considers recall, and better represents the ability of M-FAIL to identify either of the classes.

Considering F1, the best model to identify that a given example has the properties to be correctly classified by M-BERT, is based on semantic features.

The second best model, by a distance of more than 1%, also involves semantic features, but combined with lexical features. However, the only semantic feature in the most informative features for the second best model is once again the count of matched entities according to heuristics, while lexical features in such set include less count based features than in previous experiments, although still in greater number among the top 10.

### 5.4 Strategy 2 Results

Table 4 shows the top-10 results considering the best combination between M-FAIL and M-OTHER models, considering Strategy 2, that is, a M-FAIL model predicts that BERT will fail and an M-OTHER model is activated in those situations. We will represent these combinations by m1/m2 in which m1 is an M-FAIL model or ensemble and m2 is an M-OTHER.

**Table 4.** Strategy 2 results

| M-FAIL / M-OTHER features | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| d / b+l | 79.43% | 81.28% | 81.32% | 81.26% |
| b+l / b+l | 79.77% | 81.39% | 81.44% | 81.39% |
| b+l+d / b+l | 79.79% | 81.41% | 81.46% | 81.41% |
| l / b+l | 79.69% | 81.44% | 81.48% | 81.42% |
| l+d / b+l | 79.77% | 81.48% | 81.53% | 81.47% |
| b+d / b+l | 79.80% | 81.47% | 81.53% | 81.47% |
| b / b+l | 79.80% | 81.57% | 81.63% | 81.56% |
| All / b+d | 79.56% | 81.11% | 81.14% | 81.09% |
| Majority voting / b+l | 79.77% | 81.44% | 81.48% | 81.43% |
| All / b+l | **79.85%** | 81.61% | 81.67% | **81.60%** |

Results of classifying an instance with M-BERT according to at least one M-FAIL model are not shown in Table 4, since in such setting 88.87% of the test examples are classified with M-BERT, which results in performance similar to using the standalone M-BERT on the full test set (i.e., without M-FAIL models), hence lower than shown.

For the remaining settings, both from using a single M-FAIL model or an ensemble of M-FAIL models, M-BERT is employed to classify at least 32.99% of the test examples, in any of the "all" ensemble setting, and at most 70.07%, in any setting using only the M-FAIL model based on semantic features.

### 5.5 Results According to the Labels

Just to give an idea of how the best results relate with the different labels, Table 5 shows the results of the best model (d or l+d; see Table 2) according to Strategy 1, and Table 6 shows the results of the best model (all / b+l; see Table 4) according to Strategy 2.

**Table 5.** Performance per entailment label, of the best result with Strategy 1.

| Label | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| NEUTRAL | 85.80% | 82.77% | 85.80% | 84.26% |
| ENTAILMENT | 71.56% | 72.75% | 71.56% | 72.15% |
| CONTRADICTION | 79.35% | 89.26% | 79.35% | 84.01% |

**Table 6.** Performance per entailment label, of the best result with Strategy 2.

| Label | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| NEUTRAL | 86.66% | 83.58% | 86.66% | 85.09% |
| ENTAILMENT | 72.27% | 75.06% | 72.27% | 73.64% |
| CONTRADICTION | 80.62% | 86.84% | 80.62% | 83.61% |

In both cases, entailment relation is the most difficult to identify.

## 6 Conclusion and Future Work

We have presented several classifiers that perform NLI. Along with state-of-the-art BERT, other features were considered. We also implemented a model that tries to predict when BERT will fail. Various experiments here presented suggest that our semantic features are able to improve results, for instance in distinguishing ENTAILMENT from CONTRADICTIONS, as seen in results for Strategy 1. Moreover, we presented data analysis and manipulation techniques to better leverage a corpus for supervision of our models, and a novel approach to assess NLI by training a classifier to predict when a typically successful model might fail.

Machine learning in our experiments was based on linear SVM, to achieve the best performance for the least computation time and resources. However, as future work, we plan to experiment with non linear kernels, and other machine learning algorithms, such as decision trees or an ensemble of different models.

Our setup is adaptable to other corpora or features, but human supervision is required on balancing the training data and building the FAIL-CORPUS,

to prevent extreme cases on particular corpora, for instance an empty FAIL-CORPUS due to sucess of M-BERT. As such, future work also includes assessing the performance of our strategies in other corpora, and inspection of models with low performance, such as the M-FAIL models, by example analysis.

**Acknowledgements.** This work was supported by national funds through FCT, Fundação para a Ciência e Tecnologia, under project UIDB/50021/2020 and by FCT's INCoDe 2030 initiative, in the scope of the demonstration project AIA, "Apoio Inteligente a empreendedores (chatbots)", which also supports the scholarship of Pedro Fialho.

## References

1. Androutsopoulos, I., Malakasiotis, P.: A survey of paraphrasing and textual entailment methods. J. Artif. Int. Res. **38**(1), 135–187 (2010)
2. Bar-Haim, R., Dagan, I., Szpektor, I.: Benchmarking applied semantic inference: the PASCAL recognising textual entailment challenges. In: Dershowitz, N., Nissan, E. (eds.) Language, Culture, Computation. Computing - Theory and Technology. LNCS, vol. 8001, pp. 409–424. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-45321-2_19
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Trans. Assoc. Comput. Linguist. **5**, 135–146 (2017). https://doi.org/10.1162/tacl_a_00051. https://www.aclweb.org/anthology/Q17-1010
4. Bos, J.: Open-domain semantic parsing with boxer. In: Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015), pp. 301–304. Linköping University Electronic Press, Sweden, May 2015. https://www.aclweb.org/anthology/W15-1841
5. Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A large annotated corpus for learning natural language inference. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics (2015)
6. Camburu, O.M., Rocktäschel, T., Lukasiewicz, T., Blunsom, P.: e-SNLI: natural language inference with natural language explanations. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 31, pp. 9539–9549. Curran Associates, Inc. (2018)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics, Minneapolis, June 2019. https://doi.org/10.18653/v1/N19-1423
8. Fellbaum, C.: WordNet: An Electronic Lexical Database. Bradford Books (1998)
9. Fialho, P., Marques, R., Martins, B., Coheur, L., Quaresma, P.: Inesc-id@assin: Medição de similaridade semântica e reconhecimento de inferência textual. Linguamática **8**(2), 33–42 (2016). https://www.linguamatica.com/index.php/linguamatica/article/view/v8n2-4

10. Jiang, N., de Marneffe, M.C.: Evaluating BERT for natural language inference: a case study on the CommitmentBank. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 6086–6091. Association for Computational Linguistics, Hong Kong, November 2019. https://doi.org/10.18653/v1/D19-1630, https://www.aclweb.org/anthology/D19-1630
11. Kamp, H., Reyle, U.: From Discourse to Logic. Introduction to Model theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory. Kluwer, Dordrecht (1993)
12. Marelli, M., Bentivogli, L., Baroni, M., Bernardi, R., Menini, S., Zamparelli, R.: SemEval-2014 task 1: evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 1–8. Association for Computational Linguistics, Dublin, August 2014. https://doi.org/10.3115/v1/S14-2001. https://www.aclweb.org/anthology/S14-2001
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: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), pp. 216–223. European Languages Resources Association (ELRA), Reykjavik, May 2014
14. McCoy, T., Pavlick, E., Linzen, T.: Right for the wrong reasons: diagnosing syntactic heuristics in natural language inference. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 3428–3448. Association for Computational Linguistics, Florence, July 2019. https://doi.org/10.18653/v1/P19-1334, https://www.aclweb.org/anthology/P19-1334
15. Niculescu-Mizil, A., Caruana, R.: Predicting good probabilities with supervised learning. In: Proceedings of the 22nd International Conference on Machine Learning. pp. 625–632. ICML 2005, Association for Computing Machinery, New York (2005). https://doi.org/10.1145/1102351.1102430
16. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318. Association for Computational Linguistics, Philadelphia, July 2002. https://doi.org/10.3115/1073083.1073135, http://www.aclweb.org/anthology/P02-1040
17. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
18. Talman, A., Chatzikyriakidis, S.: Testing the generalization power of neural network models across NLI benchmarks. In: Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pp. 85–94. Association for Computational Linguistics, Florence, August 2019. https://doi.org/10.18653/v1/W19-4810, https://www.aclweb.org/anthology/W19-4810
19. Wang, A., et al.: Superglue: a stickier benchmark for general-purpose language understanding systems. In: Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 32, pp. 3266–3280. Curran Associates, Inc. (2019)

20. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.: GLUE: a multi-task benchmark and analysis platform for natural language understanding. In: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pp. 353–355. Association for Computational Linguistics, Brussels, November 2018. https://doi.org/10.18653/v1/W18-5446, https://www.aclweb.org/anthology/W18-5446
21. Williams, A., Nangia, N., Bowman, S.: A broad-coverage challenge corpus for sentence understanding through inference. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 1112–1122. Association for Computational Linguistics, New Orleans, June 2018. https://doi.org/10.18653/v1/N18-1101, https://www.aclweb.org/anthology/N18-1101
22. Zhang, Z., Wu, Y., Li, Z., Zhao, H.: Explicit contextual semantics for text comprehension. In: Proceedings of the 33rd Pacific Asia Conference on Language, Information and Computation (PACLIC 33) (2019)