# Team DiSaster at SemEval-2020 Task 11: Combining BERT and hand-crafted Features for Identifying Propaganda Techniques in News Media

**Anders Friis Kaas**
IT University of Copenhagen
Rued Langgaards Vej 7
2300 Copenhagen
anfk@itu.dk

**Viktor Torp Thomsen**
IT University of Copenhagen
Rued Langgaards Vej 7
2300 Copenhagen
vikt@itu.dk

**Barbara Plank**
IT University of Copenhagen
Rued Langgaards Vej 7
2300 Copenhagen
bapl@itu.dk

## Abstract

The identification of communication techniques in news articles such as propaganda is important, as such techniques can influence the opinions of large numbers of people. Most work so far focused on the identification at the news article level. Recently, a new dataset and shared task has been proposed for the identification of propaganda techniques at the finer-grained span level. This paper describes our system submission to the subtask of technique classification (TC) for the SemEval 2020 shared task on detection of propaganda techniques in news articles. We propose a method of combining neural BERT representations with hand-crafted features via stacked generalization. Our model has the added advantage that it combines the power of contextual representations from BERT with simple span-based and article-based global features. We present an ablation study which shows that even though BERT representations are very powerful also for this task, BERT still benefits from being combined with carefully designed task-specific features.

## 1 Introduction

The purpose of propaganda is to use communication to foster predetermined agendas, or to achieve a response that furthers a desired outcome (Jowett and O'Donnell, 2018).

Prior research has focused on creating machine learning models that label whole news articles or even entire news outlets as propagandistic (Rashkin et al., 2017; Barrón-Cedeño et al., 2019). To increase the granularity of these coarse models, a new data set was developed in a study by Da San Martino et al (2019), which enabled models to jointly identify fragments of propaganda within a document, while also classifying their respective propaganda techniques (Da San Martino et al., 2019; Yu et al., 2019).

This paper presents our solution (DiSaster, finishing at 11th place) to the technique classification (TC) sub-task at SemEval 2020 task 11 "Detection of propaganda techniques in news articles".[1] TC is a multi class classification problem in which a system needs to identify the propaganda techniques of a given span of an article. For instance, when given the span "stupid and petty" the system should classify it as `Loaded_Language`. Our system is an ensemble model based on stacked generalization (Wolpert, 1992) which enables the incorporation of both traditional engineered features (Nalini and Sheela, 2014) and the Transformer (Vaswani et al., 2017) based language model BERT (Devlin et al., 2019).

## 2 Related work

In addition to formulating the original problem of fine-grained propaganda identification and creating the corpus needed to solve the task, Da San Martino et al. (2019) also designed a multi-granularity neural network. This model outperformed several strong BERT baseline models in the high granularity fragment-level classification by using information from low granularity classification (e.g. document-level) to drive higher-granularity classification (e.g. paragraph-level).

As the TC sub-task of this competition does not require span detection, a multi-granularity approach is not necessary. Instead, our model is inspired by a project by Zhang and Li (2019), in which they outperformed BERT baseline models by combining a BERT model with linguistic features.



[1] All code for replication are publicly available at https://github.com/ViktorTorp/SemEval2020-TC



*Proceedings of the 14th International Workshop on Semantic Evaluation*, pages 1817–1822
Barcelona, Spain (Online), December 12, 2020.

| label | id | support | % w. 1word | Avg #words | Avg one_word_counter | Avg span_sentence_counter |
|---|---|---|---|---|---|---|
| Loaded_Language | 8 | **2123** | 24.78 | 3.82 (±4.0) | 1.65 (±2.0) | 0.8 (±1.0) |
| Name_Calling,Labeling | 9 | 1058 | 11.25 | 3.93 (±3.0) | 1.62 (±2.0) | 0.62 (±1.0) |
| Repetition | 10 | 621 | **43.64** | 2.81 (±3.0) | **6.92** (±5.0) | **1.35** (±2.0) |
| Doubt | 5 | 493 | 1.42 | 21.14 (±16.0) | 1.0 (±1.0) | 0.13 (±0.0) |
| Exaggeration,Minimisation | 6 | 466 | 6.22 | 7.44 (±6.0) | 0.97 (±0.0) | 0.63 (±1.0) |
| Appeal_to_fear-prejudice | 1 | 294 | 3.06 | 17.05 (±13.0) | 1.44 (±1.0) | 0.32 (±0.0) |
| Flag-Waving | 7 | 229 | 11.79 | 10.63 (±12.0) | 4.33 (±3.0) | 0.22 (±1.0) |
| Causal_Oversimplification | 4 | 209 | 0.0 | 21.52 (±13.0) | - (-) | 0.1 (±0.0) |
| Appeal_to_Authority | 0 | 144 | 0.0 | **23.2** (±22.0) | - (-) | 0.22 (±0.0) |
| Slogans | 11 | 129 | 6.2 | 4.33 (±3.0) | 1.25 (±2.0) | 0.23 (±1.0) |
| Whataboutism,Straw_Men,Red_Herring | 13 | 108 | 3.7 | 16.5 (±11.0) | 2.25 (±1.0) | 0.12 (±0.0) |
| Black-and-White_Fallacy | 3 | 107 | 0.0 | 18.71 (±13.0) | - (-) | 0.15 (±0.0) |
| Thought-terminating_Cliches | 12 | 76 | 1.32 | 6.13 (±4.0) | 0.0 (±0.0) | 0.27 (±0.0) |
| Bandwagon,Reductio_ad_hitlerum | 2 | 72 | 0.0 | 16.44 (±12.0) | - (-) | 0.1 (±0.0) |

Table 1: Overview of the provided training data for the SemEval 2020 Task 11 competition. The definition of % w. 1word, Avg #words, Avg one_word_counter and Avg span_span_sentence_counter are described in Section 4.2. The bold font indicates the highest value within a column. In the columns Avg #words, Avg one_word_counter and Avg span_sentence_counter are the standard deviations included in the parentheses.

## 3 Data

The provided training data for this competition contains 371 articles in which all fragments of propaganda are annotated with one of the 18 different propaganda techniques described in Da San Martino et al. (2019). However, due to a low frequency of some of the techniques, similar underrepresented techniques were merged into a superclass, while one of the techniques was eliminated completely. Thus, the TC task was a 14-class classification problem, where two of the classes were superclasses representing several techniques each (Da San Martino et al., 2020). Table 1 contains a list of all the labels along with their respective IDs that we defined.

The class distribution in the training data was very skewed (as the support in Table 1 shows); four of the labels accounted for more than 70% of the training data. As the score for the competition was calculated as the micro-average F1 over all the labels, it was crucial to get good predictions on these four classes. In order to create hand-crafted features that would increase the model's performance for these techniques, we performed a thorough data analysis whose main results are summarized in Table 1.

Table 1 shows that there is a considerable spread in the average number of words per span among the different techniques. In particular, the spans from the `Repetition` category were much shorter than other techniques, and more than 40% of its spans only contained a single word. By examining the instances of `Repetition` in which the span only contained a single word, we found that the Porter stemmed version of the word (Porter and others, 1980) often occurred several times within the article.[2] This effect is displayed as Avg one_word_counter in Table 1, which is the average number of times the stem of single word span occurs within an article. However, this value cannot be calculated for classes in which every span contains more than one word. Furthermore, a similar effect for `Repetition` was discovered when spans with more than one word were examined. The average number of times an entire span with more than one word was repeated within an article was generally much higher for `Repetition` than any other technique. This effect is shown in Table 1 as Avg span_sentence_counter. Additionally, we found that if a label was in an article, there was a much higher probability of finding another span with the same label elsewhere in the article (see Figure 2).

## 4 Model overview

Our simplest baseline model worked by always predicting highest prior probability. The label with the highest prior probability in the training set was `Loaded_Language`. We compare the results of this baseline model with our final model in Section 5, Table 2.

We tackle the problem of propaganda technique identification as a classification task where we combine

[2]E.g. in article 699291100 the stem of the word "threatened" (i.e. "threaten") was repeated 3 times

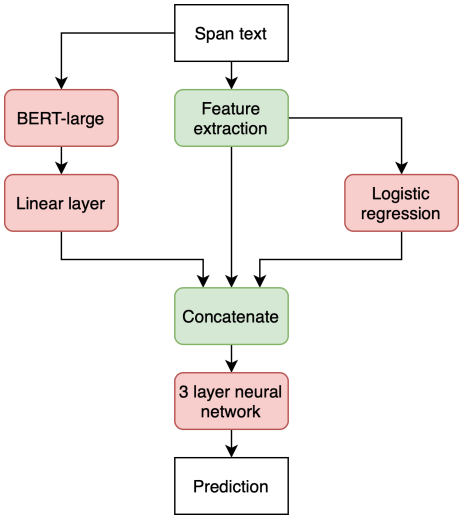


Figure 1: The full model pipeline.


Figure 2: Normalized matrix of labels co-occurring in the same article. The diagonal line indicates that articles tend to contain spans with the same label. The IDs correspond to the classes listed in Table 1.

three components using stacked generalization (Wolpert, 1992). Our model is illustrated in Figure 1 and consists of: (1) a contextualized embedding representation of the span using BERT, (2) hand-crafted features extracted from both the span and the global article structure, and (3) the scores of a traditional logistic regression model trained on the hand-crafted features. These components are combined using a feed-forward neural network as the topmost stacking classifier. All the components are described in the next subsections.

### 4.1 BERT fine-tuning

The BERT component of the pipeline consists of BERT-large with a single linear layer on top of the output, similar to the approach described in Devlin et al. (2019). This component was only used on a span-level (i.e. the actual propaganda fragments), in order to get a 14 dimensional vector of logits corresponding to the 14 propaganda technique classes.

To obtain the logits from BERT for all of the training set spans, a 10-fold stratified learning strategy was used: 10 stratified train/test splits were created from the training set. BERT-large was then initialized and fine-tuned, as suggested by Devlin et al. (2019), on each of the 10 training sets and made to predict the logits for the corresponding test sets. This method insured that logits were predicted on the whole training set without predicting on data that it was trained on. The logits for the development and test sets were created by fine-tuning on a stratified 90% sub-set of the training data and stopping early when the loss of the remaining 10% stopped decreasing.

BERT was optimized on the cross-entropy loss using the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of $2 \times 10^{-5}$ and an epsilon value of $1 \times 10^{-8}$ .

### 4.2 Feature extraction

In addition to the BERT logits, we extracted several additional features from the data. In total we extracted 54 features.[3] We found that the following five improved the performance of the model the most:

- If the span is only one word, *article_one_word_counter* (aowc) is a count of how many times the Porter stem of that word appeared in the article. Otherwise it is 0.

[3]All features are described in our GitHub repository `https://github.com/ViktorTorp/SemEval2020-TC`

- If the span is more than one word long, *article_span_sentence_counter* (assc) is a count of how many times that span appeared elsewhere in the article. Otherwise it is 0.
- *span_word_length* (swl) is a count of the number of words in the span.
- *word_count_span_sent* (wcss) is the number times that a span appears within the sentence it is presented in. E.g. the span "fake news" appears twice in the sentence "it is fake news about a fake news story."
- *word_resemble_factor* (wrf) is the inverse uniqueness of words in a span and is calculated as $\frac{\text{number of words in span}}{\text{number of unique words in span}}$.

Furthermore, a logistic regression was performed over the hand-crafted features alone using a similar stratified learning strategy as for BERT, and the resulting 14-dimensional output was used downstream in our pipeline. We compare and discuss the importance of the features in Section 6.

### 4.3 Feed-forward network

The last component of our model is a fully connected neural network with three hidden layers each consisting of 500 neurons. The network was optimized using the AdamW (Loshchilov and Hutter, 2019) optimizer with a learning rate of $2 \cdot 10^{-5}$ and an epsilon value of $1 \cdot 10^{-8}$.

As illustrated in Figure 1, the network was fed a concatenation of BERT logits, the hand-crafted features and the output of the logistic regression over the features. The resulting output was a 14 dimensional vector of logits corresponding to the 14 propaganda classes.

### 4.4 Model performance

The most compute-intense step of our model is extracting the BERT task-specific representations for the entire dataset (as outlined in Section 4.1). Fine-tuning BERT and obtaining the representations took roughly 12 hours using the Tesla K80 GPU available on Google Colab. However, once the model is trained, new representations can be obtained in seconds. The extraction of global features is quicker, taking less than 30 minutes for both the training, development and test set on a 2017 MacBook Pro with 3,1 GHz Quad-Core Intel Core i7 processor. A clear advantage of our model is its simplicity. Once the BERT features are extracted, our stacked model can be trained in about five minutes.

## 5 Experiments

To test the importance of the different components, a feature ablation study was performed. A 10-fold stratified cross-validation was then performed on the training set and the micro-average F1 score was recorded. The models used in this experiment were implemented in Python using a PyTorch framework (Paszke et al., 2019). The features were created using a mixture of SpaCy and NLTK (Honnibal and Montani, 2017; Bird et al., 2009), whereas for BERT we use the Huggingface library (Wolf et al., 2019).

All results obtained from the ablation study are summarized in Table 2 along with the model's scores from a 10-fold cross validation on the training set. Furthermore, Table 2 also shows the final micro-average F1 score we got on the official competition development and test sets.

## 6 Discussion

As evident from the ablation study (Table 2), the most important component of our learning setup was BERT. However, as BERT is used at the span level, it is only able to predict a label based on the tokens in a given span. Due to this local behavior, BERT alone was struggling to correctly predict the `Repetition` class. This was most likely because the words or phrases that were repeated were not necessarily in the span, but spread throughout the article, which the data exploration in Section 3 also supports. However, this was a problem as `Repetition` was the third most frequent class in the training set. It was for this reason that we decided to extract and use additional global (article level) features from the data set. The most important extracted feature for `Repetition`, was the *article_one_word_counter*. This features directly tell the final neural network if a word has been repeated in the article and removing this global feature shows its importance, as the f1 score for the class `Repetition` drops from 0.646 to

| | | | Feature ablation from the full model | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | baseline | Full model | - BERT | - LR | - HCF | - HCF & - LR | - Finetuning | - wrf | - aowc | - assc | - swl | - wcss |
| Cross validation training set | 0.346 | **0.672** | 0.443 | 0.669 | 0.668 | 0.667 | 0.467 | 0.671 | 0.670 | 0.669 | 0.667 | 0.671 |
| Development set | 0.306 | **0.628** | 0.436 | 0.615 | 0.598 | 0.591 | 0.302 | 0.617 | 0.604 | 0.609 | 0.618 | 0.619 |
| Test set | - | **0.566** | - | - | - | - | - | - | - | - | - | - |
| Appeal_to_Authority | 0.000 | 0.341 | 0.014 | 0.316 | 0.308 | 0.304 | 0.049 | 0.342 | **0.357** | 0.323 | 0.293 | 0.351 |
| Appeal_to_fear-prejudice | 0.000 | 0.447 | 0.019 | 0.462 | 0.442 | 0.452 | 0.163 | **0.475** | 0.455 | 0.472 | 0.448 | 0.456 |
| Bandwagon,Reductio_ad_hitlerum | 0.000 | 0.162 | 0.000 | 0.158 | 0.204 | **0.214** | 0.000 | 0.152 | 0.168 | 0.204 | 0.160 | 0.174 |
| Black-and-White_Fallacy | 0.000 | 0.200 | 0.000 | 0.198 | 0.279 | 0.255 | 0.000 | 0.190 | 0.236 | 0.173 | **0.281** | 0.240 |
| Causal_Oversimplification | 0.000 | 0.433 | 0.000 | 0.405 | 0.424 | 0.441 | 0.056 | 0.425 | **0.456** | 0.434 | 0.451 | 0.437 |
| Doubt | 0.000 | 0.640 | 0.384 | 0.642 | 0.639 | 0.633 | 0.426 | 0.643 | **0.650** | 0.638 | 0.636 | 0.636 |
| Exaggeration,Minimisation | 0.000 | 0.530 | 0.000 | 0.535 | 0.532 | **0.546** | 0.152 | 0.531 | 0.525 | 0.524 | 0.526 | 0.528 |
| Flag-Waving | 0.000 | 0.622 | 0.000 | 0.612 | 0.607 | 0.606 | 0.373 | 0.617 | **0.633** | 0.620 | 0.604 | 0.622 |
| Loaded_Language | 0.515 | **0.796** | 0.630 | 0.793 | **0.796** | 0.793 | 0.634 | 0.793 | 0.790 | 0.789 | 0.790 | **0.796** |
| Name_Calling,Labeling | 0.000 | **0.792** | 0.338 | 0.790 | 0.787 | 0.789 | 0.403 | 0.787 | 0.786 | 0.785 | 0.785 | 0.788 |
| Repetition | 0.000 | **0.646** | 0.541 | 0.638 | 0.619 | 0.615 | 0.543 | 0.638 | 0.621 | 0.634 | 0.638 | 0.638 |
| Slogans | 0.000 | 0.518 | 0.000 | 0.514 | 0.520 | **0.530** | 0.172 | 0.516 | **0.530** | 0.510 | 0.526 | 0.529 |
| Thought-terminating_Cliches | 0.000 | **0.343** | 0.000 | 0.318 | 0.341 | 0.331 | 0.000 | 0.284 | 0.321 | 0.293 | 0.321 | 0.299 |
| Whataboutism,Straw_Men,Red_Herring | 0.000 | 0.157 | 0.000 | 0.156 | 0.141 | 0.162 | 0.000 | **0.173** | 0.144 | 0.131 | 0.127 | 0.101 |

Table 2: Summary of the results and the results of the ablation study. Columns to the right of 'Full model' are ablated features. The rows in the bottom section are individual F1-scores per class from the cross-validation on the training set. The rows in the top section are the micro-averaged F1 scores on the training set, the development set and the test set. The abbreviations in the columns are: LR (logreg), HCF (hand-crafted features), wrf (word_resemble_factor), aowc, (article_one_word_counter), assc (article_span_sentence_counter), swl (span_word_length), wcss (word_count_span_sent).

0.621 (Table 2). This is also supported by our data analysis (Section 3, Table 1) which shows that the Avg one_word_counter are much higher for `Repetition` compared to the other labels.

The fact that we obtained better quality predictions by augmenting BERT-predictions with additional information about the text shows that feature engineering is still a relevant discipline as other recent research also suggests (Wu et al., 2018; Zhang and Li, 2019).

The augmented BERT approach worked well on both the training set and the development set, but our score dropped significantly when predicting on the test set (0.628 dev set micro F1 $\rightarrow$ 0.566 test set micro F1). As we do not have access to the test set labels, a detailed error analysis is difficult for now and left for future work. However, by comparing the F1 score from the official test set with the cross validation scores in Table 2, we do see a particularly large drop in F1 for the `Repetition` category (from 0.646 cross validation $\rightarrow$ 0.204 test). This drop in `Repetition` F1 can also be observed for the other participants in the competition. This may be due to overfitting the model to the training and development sets. It may also be due to the test set having a slightly different distribution than the training and development sets.

Finally, we explored several approaches to exploit the phenomenon of labels co-occurring in articles (Section 3, Figure 2), namely using an RNN and an attention-based model over all the spans in an article. Additionally we tried feeding a neural network the average BERT prediction for all spans in an article in addition to the other features we included. Unfortunately, we were unable to improve the performance using these approaches.

## 7 Conclusion

In this paper we have presented the model we used in the SemEval 2020 competition, Task 11: "Detection of Propaganda Techniques in News Articles". The model consists of several components, the most important one being the BERT component. We combined BERT with valuable global and local features extracted from the articles and spans which improved the predictive power of our model, especially in the `Repetition` category. We ended up with a micro average F1 score of 0.56648 on the official test set, earning us an 11th place (out of 32 teams) overall in the competition.

As visualized in Figure 2, the labels were not distributed uniformly throughout the articles. In particular, if a technique was used in an article, there was a much higher chance than expected of finding it elsewhere in the article. We still believe that the model can be improved by including new features that contain information about the different labels' trends. Furthermore, we would like to exploit the tendency that a label within an article has a higher chance of occurring later in the same article.

## References


Alberto Barrón-Cedeño, Giovanni Da San Martino, Israa Jaradat, and Preslav Nakov. 2019. Proppy: A system to unmask propaganda in online news. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:9847–9848, Jul.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly, Beijing.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news articles. 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 2019*, Hong Kong, China, November.

Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. SemEval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, SemEval 2020, Barcelona, Spain, September.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

G.S. Jowett and V. O'Donnell. 2018. *Propaganda & Persuasion*. SAGE Publications.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. Ernest N. Morial Convention Center, New Orleans, January.

K. Nalini and Dr. L. Jaba Sheela. 2014. Survey on text classification. *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, 1, Jul.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Martin F Porter et al. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937, Copenhagen, Denmark, September. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NeurIPS 2017, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

David Wolpert. 1992. Stacked generalization. *Neural Networks*, 5:241–259, 12.

Minghao Wu, Fei Liu, and Trevor Cohn. 2018. Evaluating the utility of hand-crafted features in sequence labelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2850–2856, Brussels, Belgium, October-November. Association for Computational Linguistics.

Seunghak Yu, Giovanni Da San Martino, and Preslav Nakov. 2019. Experiments in detecting persuasion techniques in the news. *ArXiv*, abs/1911.06815.

Yue Zhang and Jiawei Li. 2019. The death of feature engineering? —— bert with linguistic features on squad 2.0. Technical Report CS224n, Stanford University.