Assignment 1

Fabio Zanotti, Antonio Morelli, Federico Rullo, and **Edoardo Conca**
Master's Degree in Artificial Intelligence, University of Bologna
{ fabio.zanotti, antonio.morelli, federico.rullo, edoardo.conca }@studio.unibo.it

Abstract

The abstract is a very brief summary of your report. Try to keep it no longer than 15-20 lines at most. Write your objective, your approach, and your main observations (what are the findings that make this report worthwhile reading?)

1 Introduction

The Assignment consisted in implementing a Part Of Speech Tagging system using neural architectures, given a corpus of documents, the objective was to predict the Part Of Speech tag for each word. The system has been implemented using 3 different models, and comparing them in order to see which one would perform better on the given data. The Three different models are:

- Baseline - consisting of a single Bidirectional LSTM layer;
- Gated Recurrent Units (GRU);
- Baseline with an additional Bidirectional LSTM layer.

Bidirectional LSTM layers are able to process sequential data in both forward and backward directions, this allows the model to capture contextual information from both past and future. This is particularly useful for natural language processing tasks as the meaning of words can sometimes depend on the context in which they are used, one advantage of using it for POS tagging is that it allows one to predict the tag for each word in a sentence simultaneously, this is particularly useful when dealing with long sentences. On the other hand Gated Recurrent Units are a type of recurrent neural networks that are often used in natural language processing tasks such as **Part of Speech** tagging, they are similar to LSTM networks but have a simpler structure and fewer parameters, Gru layers can be used to process a sequence of words and predict its POS tag for each word in the sequence. Our approach was to implement these different models and compare them based on the average Macro-F1 score which is a common metric used to evaluate the performance of models and was also the requested metric.

MAX 1 COLUMN FOR ASSIGNMENT REPORTS / 2 COLUMNS FOR PROJECT OR PW / 3 FOR COMBINED REPORTS.

Then give a short overview of known/standard-/possible approaches to that problems, if any, and what are their advantages/limitations.

After that, discuss your approach, and motivate why you follow that approach. If you are drawing inspiration from an existing model, study, paper, textbook example, challenge, . . . , be sure to add all the necessary references (Chowdhery et al., 2022; Lorenzo et al., 2022; Antici et al., 2021; Nakov et al., 2021; Röttger et al., 2022; Lippi and Torroni, 2016).[1]

Next, give a brief summary of your experimental setup: how many experiments did you run on which dataset. Last, make a list of the main results or take-home lessons from your work.

2 System description

MAX 1 COLUMN FOR ASSIGNMENT REPORTS / 4 COLUMNS FOR PROJECT OR PW / 6 FOR COMBINED REPORTS.

We initially imported the dataset from the Natural Language Toolkit(NLTK) library, which we pre-processed. We removed the "-NONE-" tag, this way the model will have fewer examples of unstructured data to learn, and can focus on the examples that are most relevant to the POS tagging increasing the accuracy of the results. After pre-processing we build the vocabulary, for this, we used GloVe (Global Vectors for Word Representation) which is a method for learning vector representations of words called

[1] Add only what is relevant.

"word embeddings" from a large corpus of text. Because some words may not be present in the GloVe vocabulary we expanded it by integrating Out Of Vocabulary words which are present in the dataset. Since there are different versions of GloVe, each of which includes an increasing number of dimensions we settled for the smaller one, 50-dimensional, which is easier to work with and computationally more efficient. By using GloVe embedding to set the initial weights of the model we could take advantage of the pre-trained word representations and fine-tune them to suit our task. Finally, we implemented the three models for the POS Tagging system. All these models have the same input and output layers, we used Glove Embedding as an input layer and TimeDistributed as an output layer. The models differ in the hidden layers, the first model, which is the Baseline model, is composed of only one Bidirectional LSTM hidden layer. The Second model has a GRU hidden layer. The third mdoel is the Baseline model extended with an additional Bidirectional LSTM layer. And the fourth and final model is the Baseline with an additional Dense layer.

Describe the system or systems you have implemented (architectures, pipelines, etc), and used to run your experiments. If you reuse parts of code written by others, be sure to make very clear your original contribution in terms of

- architecture: is the architecture your design or did you take it from somewhere else
- coding: which parts of code are original or heavily adapted? adapted from existing sources? taken from external sources with minimal adaptations?

It is a good idea to add figures to illustrate your pipeline and/or architecture(s) (see Figure 5)

3 Experimental setup and results

Describe how you set up your experiments: which architectures/configurations you used, which hyperparameters and what methods were used to set them, which optimizers, metrics, etc.
Then, **use tables** to summarize your findings (numerical results) in validation and test. If you don't have experience with tables in LaTeX, you might want to use LaTeXtable generator to quickly create a table template.

4 Discussion

MAX 1.5 COLUMNS FOR ASSIGNMENT REPORTS / 3 COLUMNS FOR PROJECT / 4 FOR COMBINED REPORTS. ADDITIONAL EXAMPLES COULD BE PLACED IN AN APPENDIX AFTER THE REFERENCES IF THEY DO NOT FIT HERE.

Here you should make your analysis of the results you obtained in your experiments. Your discussion should be structured in two parts:

- discussion of quantitative results (based on the metrics you have identified earlier; compare with baselines);
- error analysis: show some examples of odd-/wrong/unwanted outputs; reason about why you are getting those results, elaborate on what could/should be changed in future developments of this work.

5 Conclusion

MAX 1 COLUMN.

In one or two paragraphs, recap your work and main results. What did you observe? Did all go according to expectations? Was there anything surprising or worthwhile mentioning? After that, discuss the main limitations of the solution you have implemented, and indicate promising directions for future improvement.

6 Links to external resources

THIS SECTION IS OPTIONAL

Insert here:

- a link to your GitHub or any other public repo where one can find your code (only if you did not submit your code on Virtuale);
- a link to your dataset (only for non-standard projects or project works).

DO NOT INSERT CODE IN THIS REPORT

References

Francesco Antici, Luca Bolognini, Matteo Antonio Inajetovic, Bogdan Ivasiuk, Andrea Galassi, and Federico Ruggeri. 2021. SubjectivITA: An Italian corpus for subjectivity detection in newspapers. In *CLEF*, volume 12880 of *Lecture Notes in Computer Science*, pages 40–52. Springer.

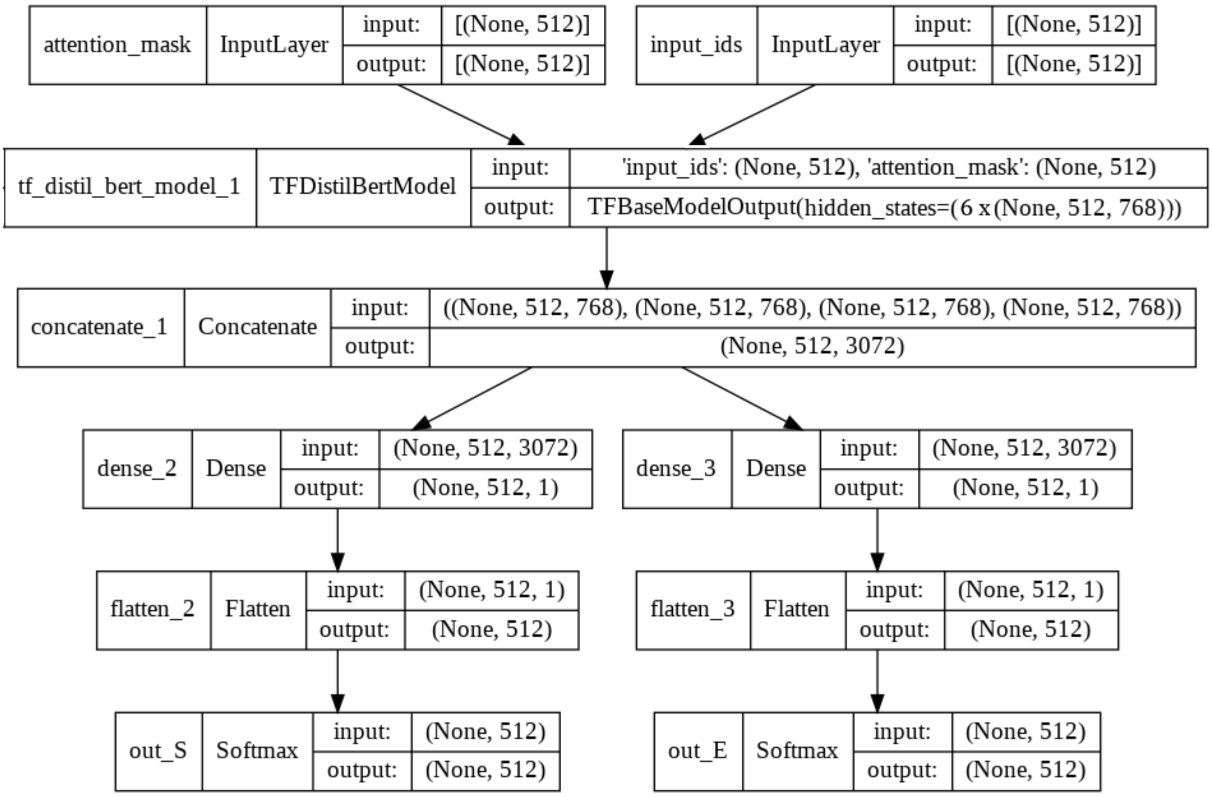
attention_mask | InputLayer | input: | [(None, 512)] | output: | [(None, 512)]
input_ids | InputLayer | input: | [(None, 512)] | output: | [(None, 512)]
tf_distil_bert_model_1 | TFDistilBertModel | input: | 'input_ids': (None, 512), 'attention_mask': (None, 512) | output: | TFBaseModelOutput(hidden_states=(6 x (None, 512, 768)))
concatenate_1 | Concatenate | input: | ((None, 512, 768), (None, 512, 768), (None, 512, 768), (None, 512, 768)) | output: | (None, 512, 3072)
dense_2 | Dense | input: | (None, 512, 3072) | output: | (None, 512, 1)
dense_3 | Dense | input: | (None, 512, 3072) | output: | (None, 512, 1)
flatten_2 | Flatten | input: | (None, 512, 1) | output: | (None, 512)
flatten_3 | Flatten | input: | (None, 512, 1) | output: | (None, 512)
out_S | Softmax | input: | (None, 512) | output: | (None, 512)
out_E | Softmax | input: | (None, 512) | output: | (None, 512)

Figure 1: Baseline architecture

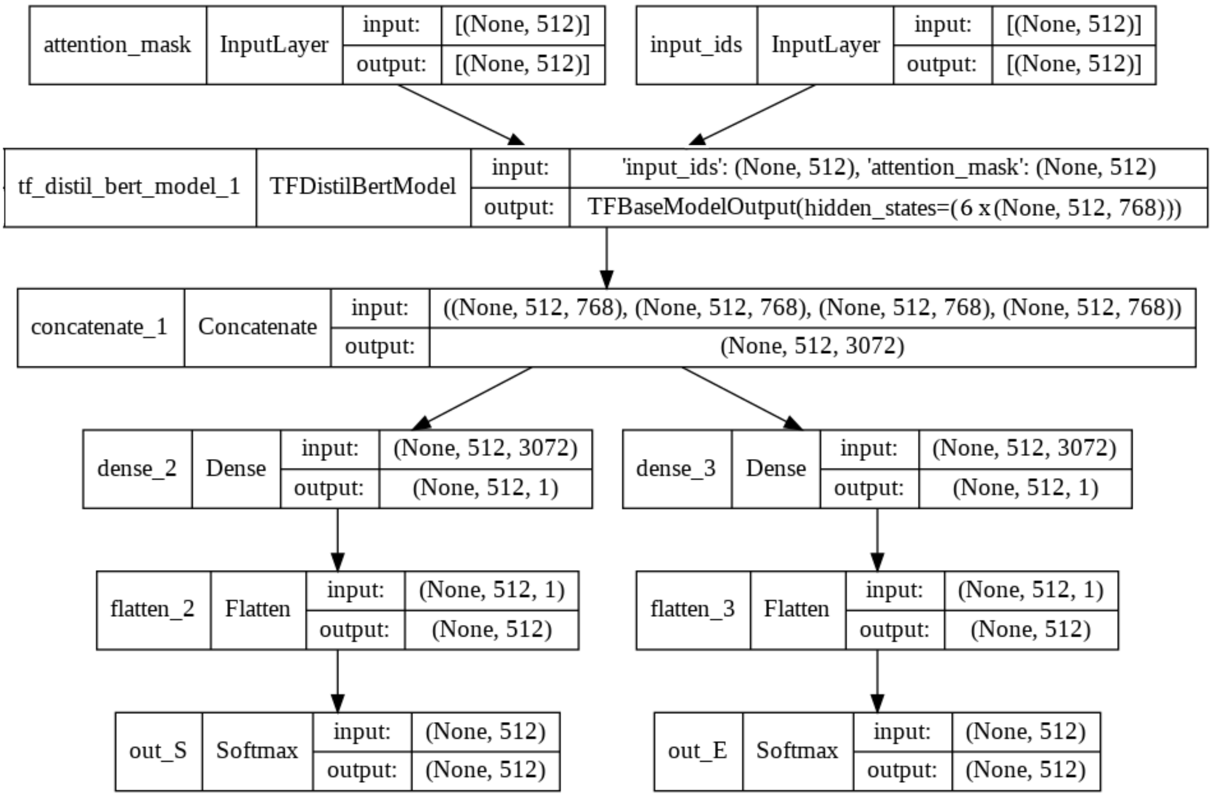
attention_mask | InputLayer | input: | [(None, 512)] | output: | [(None, 512)]
input_ids | InputLayer | input: | [(None, 512)] | output: | [(None, 512)]
tf_distil_bert_model_1 | TFDistilBertModel | input: | 'input_ids': (None, 512), 'attention_mask': (None, 512) | output: | TFBaseModelOutput(hidden_states=(6 x (None, 512, 768)))
concatenate_1 | Concatenate | input: | ((None, 512, 768), (None, 512, 768), (None, 512, 768), (None, 512, 768)) | output: | (None, 512, 3072)
dense_2 | Dense | input: | (None, 512, 3072) | output: | (None, 512, 1)
dense_3 | Dense | input: | (None, 512, 3072) | output: | (None, 512, 1)
flatten_2 | Flatten | input: | (None, 512, 1) | output: | (None, 512)
flatten_3 | Flatten | input: | (None, 512, 1) | output: | (None, 512)
out_S | Softmax | input: | (None, 512) | output: | (None, 512)
out_E | Softmax | input: | (None, 512) | output: | (None, 512)

Figure 2: GRU architecture

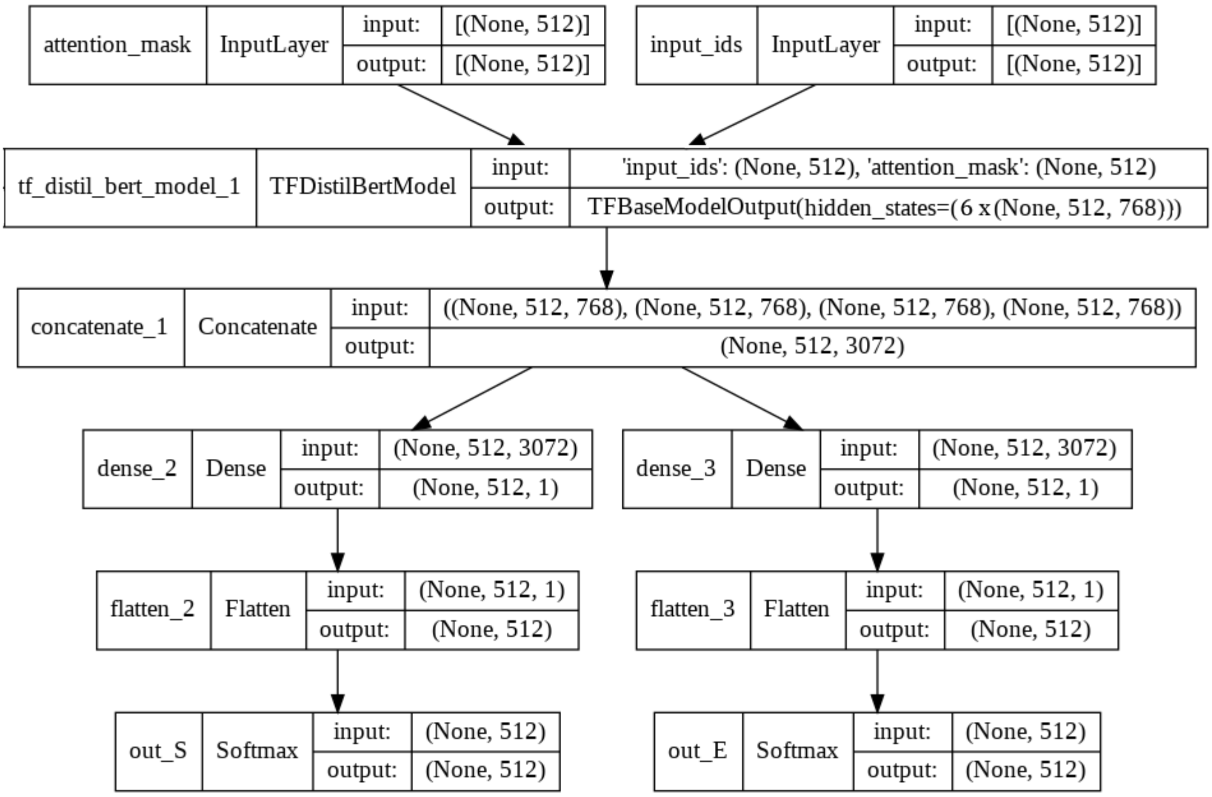

Figure 3: Additional LSTM architecture

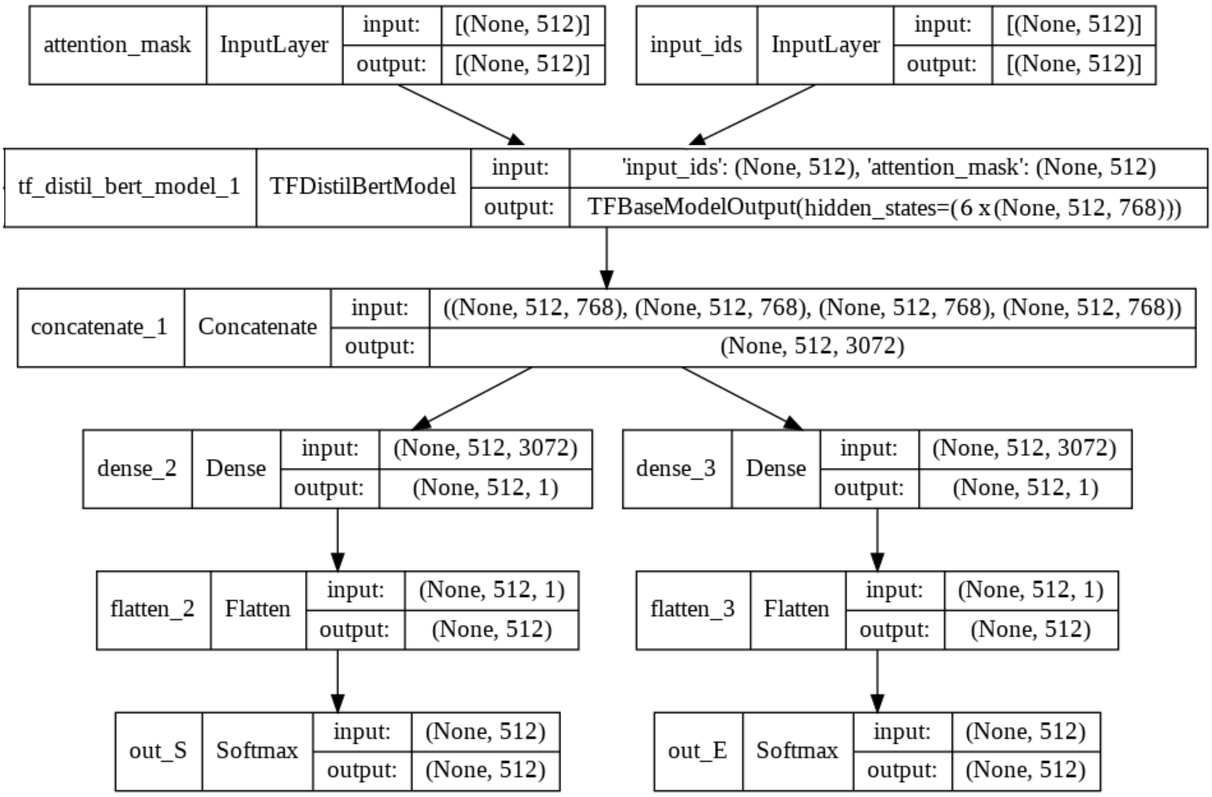

Figure 4: Additional Dense architecture

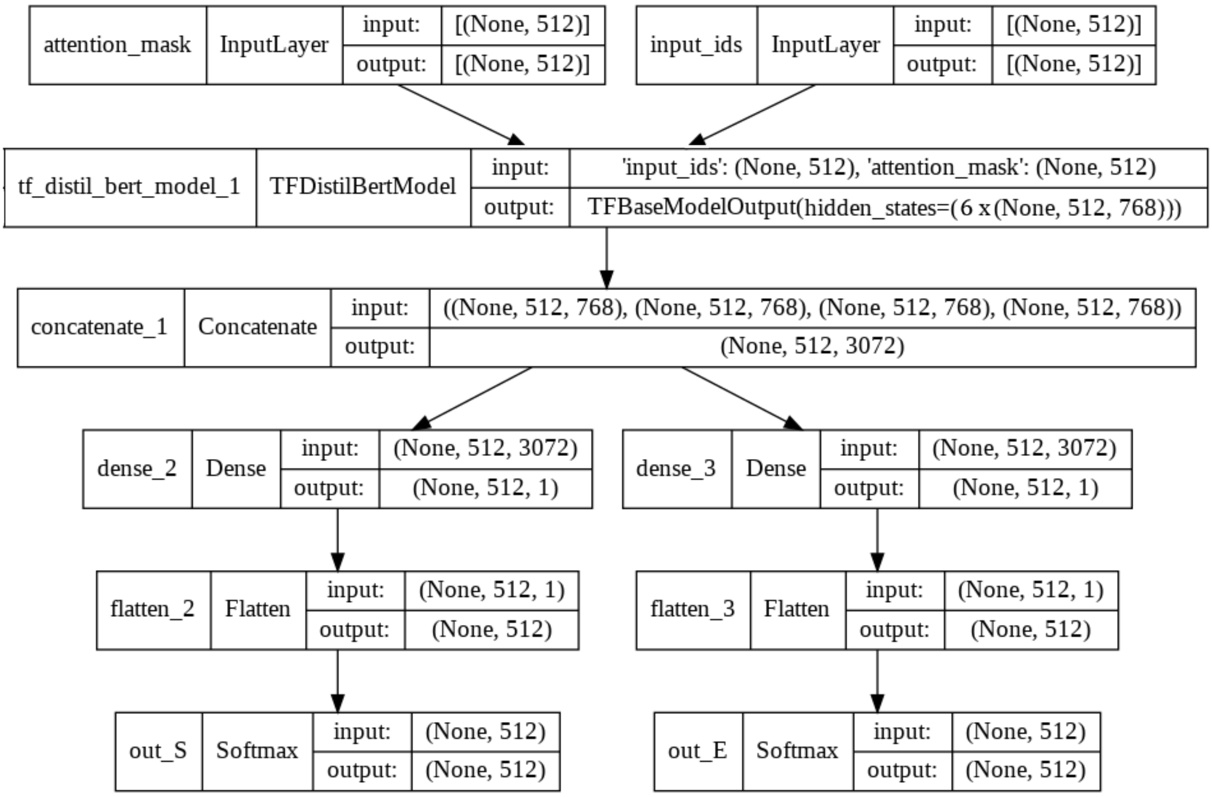

Figure 5: Model architecture

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling language modeling with pathways. *CoRR*, abs/2204.02311.

Marco Lippi and Paolo Torroni. 2016. Argumentation mining: State of the art and emerging trends. *ACM Transactions on Internet Technology*, 16(2):10:1–10:25.

Abelardo Carlos Martinez Lorenzo, Marco Maru, and Roberto Navigli. 2022. Fully-semantic parsing and generation: the BabelNet meaning representation. In *ACL (1)*, pages 1727–1741. Association for Computational Linguistics.

Preslav Nakov, David P. A. Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, Alberto Barrón-Cedeño, Paolo Papotti, Shaden Shaar, and Giovanni Da San Martino. 2021. Automated fact-checking for assisting human fact-checkers. In *IJCAI*, pages 4551–4558. ijcai.org.

Paul Röttger, Bertie Vidgen, Dirk Hovy, and Janet B. Pierrehumbert. 2022. Two contrasting data annotation paradigms for subjective NLP tasks. In *NAACL-HLT*, pages 175–190. Association for Computational Linguistics.