LEVERAGING PRE-TRAINED LANGUAGE MODELS FOR KEY POINT MATCHING

Manav Nitin Kapadnis, Sohan Patnaik, Siba Smarak Panigrahi, Varun Madhavan, Abhilash Nandy

IIT Kharagpur

Shared Task Description

The Quantitative Summarization - Key Point Analysis (KPA) Shared Task requires participants to identify the keypoints in a given corpus.

In track 1, given a debatable topic, a set of keypoints per stance, and a set of crowd arguments supporting or contesting the topic, participants must report for each argument the corresponding match score for each keypoint under the same stance towards the topic.

Dataset Description

The ArgKP-2021 dataset [1] which was the main dataset used for the shared task consists of approximately **27,520** argument/keypoint pairs for **31** controversial topics. Each of the pairs is labeled as matching or nonmatching, along with a stance towards the topic.

The train data comprises 5583 arguments and 207 keypoints, the validation data 932 arguments and 36 keypoints and the test data 723 arguments and 33 keypoints.

Argument	Keypoint	Label
Human cloning is just a medical advance more, it does not mean that we are going to create cloned human armies, that is an exaggeration	Cloning can be used for organ replication	0
	Cloning can be used to create more advanced humans	0
	Cloning helps those who can't otherwise have a child	0
	Cloning promotes health	0
	Cloning promotes science/research	1

Data Augmentation

- We further tried to experiment with sentence similarity pre-training task on two additional datasets.The two datasets used were the STS benchmark dataset and the IBM Debater® - IBM Rank 30k dataset.
- For the STS dataset, we normalized the target similarity score to bring the scores between 0 and 1. No additional preprocessing was done to the text. The two input sentences were concatenated into a single sentence and then directly fed to the model. We trained our model on STS dataset for 6 epochs and on the main dataset for 3 epochs.
- For the IBM Rank 30k dataset, we used the MACE [2] Probability score as the target column, which signifies the argument quality score for the corresponding topic. This is analogous to our approach for main task, wherein we output a similarity score for each argument-keypoint pair.

Overview of the Approach

The approach to our system revolves around the fact of using pre-trained language models (especially transformers) along with additional features such as dependency parsed features, POS tag features and Tf-idf features. We leverage the transformer based models such as BERT, BART, DeBERTa and their corresponding large versions for learning contextual representations of a keypoint - argument pair. The key-points and arguments are individually concatenated, along with the topic (in the same order) for additional context information. We then obtain the contextual representation of this triplet and concatenate to it an encoded feature vector of additional features (one of Dependency Parse based features, Parts-of-Speech based features, and Tf-idf vectors). This concatenated vector was then passed through dense layers and a sigmoid activation to get a final similarity score (within 0 and 1) between the argument and the key-point given a topic.

Model	Additional Dataset	mAP Strict	mAP Relaxed
BERT-large	STS	0.818 ± 0.045	0.933 ± 0.016
RoBERTa-large	STS	0.905 ± 0.007	$\mathbf{0.986 \pm 0.004}$
BART-large	STS	$\mathbf{0.920 \pm 0.005}$	0.967 ± 0.036
DeBERTa-large	STS	0.912 ± 0.004	0.983 ± 0.003
BERT-large	IBM Rank 30k	0.793 ± 0.029	0.914 ± 0.019
RoBERTa-large	IBM Rank 30k	0.872 ± 0.006	0.974 ± 0.003
BART-large	IBM Rank 30k	$\mathbf{0.921 \pm 0.018}$	$\mathbf{0.982 \pm 0.002}$
DeBERTa-large	IBM Rank 30k	0.894 ± 0.017	0.982 ± 0.008

Fig. 1: Results on Additional Datasets

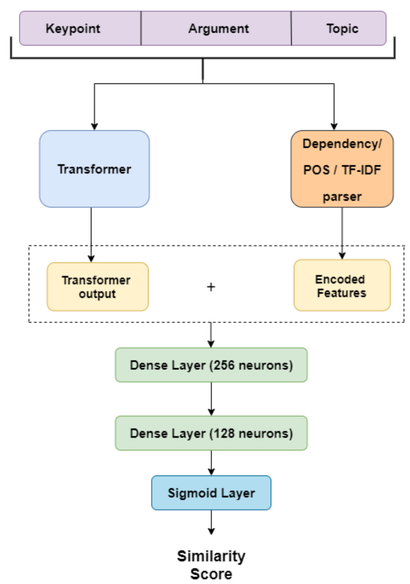

Fig. 2: Model Architecture

Extra features

As mentioned earlier, we experimented with three types of features i.e., Dependency Parsing features, Parts of Speech features and Tf-idf features.

1. **Dependency Parsing features**: In order to capture the syntactic structure of structure of the sentences, we added dependency parse tree of the sentences as an additional feature. The dependency features such as aux, amod, nsubj were label encoded according to the descending order of their occurances in the dataset. Finally, these encoded features were concatenated with the output of the transformer model and passed to the subsequent layers.
2. **Parts of Speech features**: With a similar approach to dependency parsed features, we obtained the label encoded Parts of Speech features according to descending order of their occurrences and concatenated them with the transformer output before feeding them into the fully-connected layers.
3. **Tf-idf features**: With an intuition to get better results by combining lexical overlap-based features with semantic features obtained from the transformer, we obtained Tf-idf vectors of the (key-point, argument, topic) triplet and concatenated with the transformer output before feeding them into the subsequent layers. Tf is short for Term-frequency that gives the degree of presence of a word in a triplet, whereas idf is short for inverse-document-frequency that gives the measurement of the uniqueness of a word to a particular triplet with respect to the set of all triplets in the dataset. Tf-Idf can be understood by the word frequencies weighted by the words' uniqueness to the selected dataset. A word which can be seen almost everywhere in a target triplet and cannot be found in other triplet in the corpus has high degree of uniqueness on the target triplet and hence a high Tf-Idf weight on that triplet. This feature along with DeBERTa-large model helped us achieve the best mAP strict score.

Feature	Best Model	mAP Strict	mAP Relaxed
Dep	BART-large	0.868 ± 0.023	0.977 ± 0.015
POS	BART-large	0.906 ± 0.011	0.987 ± 0.005
Tf-idf	DeBERTa-large	$\mathbf{0.911 \pm 0.005}$	$\mathbf{0.987 \pm 0.008}$

References

[1] Roni Friedman et al. "Overview of the 2021 Key Point Analysis Shared Task". In: (2021). arXiv: 2110.10577 [cs.CL].

[2] Dirk Hovy et al. "Learning Whom to Trust with MACE". In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, June 2013, pp. 1120–1130. URL: https://aclanthology.org/N13-1132.