Guidelines and template for NLP coursework report
3-cfu Project Work

Daniele Baiocco
Master's Degree in Artificial Intelligence, University of Bologna
daniele.baiocco@studio.unibo.it

Abstract

This paper explores the automatic quantification of argumentative components in abstracts from Randomized Controlled Trials (RCTs). Two distinct modeling approaches are investigated. The first is a sequence classification model that predicts token-level labels (B-Claim, I-Claim, B-Premise, I-Premise, O) using a POS tagging framework. These token-level predictions are then aggregated to derive a probability distribution across claims, premises, and outside components. The second approach directly estimates the proportions of claims, premises, and outside components by predicting directly the probability distribution. The performance of these models is evaluated using Kullback-Leibler Divergence (KLD), Mean Absolute Error (MAE), Mean Squared Error (MSE), applied to both the entire probability distribution and specifically to the claims. Results demonstrate that the "classify and count" method outperforms the direct quantification approach in accurately estimating the proportion of claims within the abstracts, highlighting its effectiveness for this dataset.

1 Introduction

2 Introduction

Argumentation Mining (AM) has emerged as a vital area of research, focusing on the automatic identification and analysis of argumentative components in text, such as claims, premises, and their interrelations. In the medical domain, particularly within Randomized Controlled Trial (RCT) abstracts, these argumentative elements play a crucial role in evidence synthesis and literature reviews, supporting clinicians and researchers in making informed, evidence-based decisions.

I used the dataset introduced by Mayer et al. (2020, 2021) (Mayer et al., 2021), which consists of more than 500 RCT abstracts annotated with 4,198 argument components, 2,601 argument relations, and 3,351 outcomes across five disease areas (neoplasm, glaucoma, hepatitis, diabetes, and hypertension). The annotations capture major claims, claims, premises and argumentative relations between these components (e.g., improved, increased, decreased, no difference, no occurrence).

The primary goal of this project is to investigate the best approach for quantifying argumentative content, specifically the amount of claims, from RCT abstracts. This quantification is valuable because RCT papers with a higher proportion of argumentative topics—such as claims and evidence—are often more relevant for literature reviews and evidence synthesis. Developing an effective predictor for argumentative content could significantly enhance the ability to identify the most impactful and relevant papers for systematic reviews and meta-analyses.

To this end, I explore two BERT-based models: one employing a sequence classification approach that aggregates token-level predictions to quantify argumentative content, and another that directly predicts the proportions of claims and premises within a text. These models are evaluated using metrics such as Kullback-Leibler Divergence, Mean Absolute Error, and Mean Squared Error to identify the most effective strategy for quantifying argumentative topics.

By leveraging the annotated dataset from Mayer et al. (Mayer et al., 2020), this research contributes to the development of intelligent systems capable of identifying high-value RCTs.

3 Background

To address the objectives of this project, inspiration was drawn from the QuaPy library, a Python-based framework dedicated to quantification (Moreo et al., 2021). QuaPy includes various quantification approaches, such as Classify & Count (CC), which is the simplest aggregative

quantifier. CC relies on the label predictions of a classifier to estimate class prevalence. This idea forms the basis of the first model in this project, which applies part-of-speech (POS) tagging to identify labels at the token level and then counts the predictions within an abstract.

Additionally, this work incorporates concepts from QuaNet, a deep-learning-based recurrent quantifier implemented in the QuaPy library, designed to predict probability distributions directly. However, while QuaPy's methods are highly effective, their implementations are primarily dataset-level, focusing on predicting class distributions over an entire dataset. For instance, given binary target classes (e.g., 0 and 1), the Classify & Count method predicts labels for all textual instances in the dataset and aggregates the counts of each class to derive a probability distribution.

In contrast, this project focuses on quantification at the instance level. Specifically, the Classify & Count approach, in my case, involves applying POS tagging to derive labels at the token level, counting occurrences of each class within individual abstracts, and generating a probability distribution for each instance rather than at the dataset level. While QuaPy could not be directly utilized in this context due to these differences in granularity, the underlying concepts and methodologies inspired the approaches developed in this work.

4 System description

For what concerns the architectures, I implemented a Classification & Count model and a Quantification model. The first one is built upon a POS Tagging model. They are explained below.

4.1 Part-of-Speech Tagging

The model used for Part-of-Speech (POS) tagging is based on the `BertForSequenceTagging` architecture. This model combines the pre-trained `BERT` model with additional components to handle the sequence tagging task effectively. The main components are as follows:

- **BERT Encoder:** The backbone of the model is the `BERT` encoder, which outputs contextualized embeddings for each token in the input sequence.
- **Bidirectional GRU:** A bidirectional Gated Recurrent Unit (GRU) processes the sequence embeddings from the BERT encoder, enabling the model to capture sequential dependencies effectively.
- **Conditional Random Field (CRF):** A CRF layer is applied on top of the GRU outputs to model the dependencies between output labels, ensuring valid label sequences.
- **Classifier:** A fully connected layer maps the GRU outputs to emission scores, which represent the logits for each label class.

The model is trained to maximize the log-likelihood of the correct label sequence using the CRF layer. Custom initialization techniques are applied to ensure optimal weight initialization, including Xavier initialization for the GRU and CRF parameters. The implementation is taken from (Mayer et al., 2021).

4.2 Classification and Counting

The classification and counting task is handled by a wrapper model, `ClassificationAndCounting`, which builds upon the `BertForSequenceTagging` architecture. The key elements of this model are:

- **Sequence Tagging Base:** The `BertForSequenceTagging` model serves as the core learner, providing token-level predictions.
- **Aggregation Mechanism:** Predictions are aggregated across the sequence to compute a probability distribution over the label classes. This is facilitated by a custom `DataProcessor` class, which computes the label distribution based on token-level predictions and attention masks.

4.3 Quantification

The quantification task employs the `BertForLabelDistribution` model, designed to estimate the label distribution of an input sequence. The architecture is structured as follows:

- **BERT Encoder:** Similar to the POS tagging model, the `BERT` encoder generates contextual embeddings for the input sequence.
- **Bidirectional GRU:** The embeddings are further processed by a bidirectional GRU to capture sequential patterns.

- **Classifier:** A fully connected layer maps the GRU outputs to logits for each label class. The logits corresponding to the CLS token are used to compute the final label distribution.
- **Softmax Layer:** A softmax activation is applied to the logits to obtain a probability distribution over the label classes.
- **Custom Loss Functions:** The model is trained using task-specific loss functions, including KLD and MAE, to measure the divergence between predicted and ground truth distributions.

5 Data

As previously mentioned, the dataset used in this study was introduced by Mayer et al. (2020, 2021) (Mayer et al., 2020). Specifically, I worked with a preprocessed version of this dataset, which can be found in (Mayer et al., 2021). The data is formatted in a CoNLL-style file and employs BIO Tagging annotations, where the labels include B-Claim, I-Claim, B-Premise, I-Premise, and O, with Major Claims being categorized as Claims. The preprocessing begins with reading input data, where each line corresponds to a token, its associated labels, and other metadata. To handle text at the sub-token level, tokenization is performed using a tokenizer capable of breaking down tokens into subwords. During this step, to ensure the alignment between tokens and labels remains intact, subword tokens inherit the label of their parent token. If a subword token is marked as part of a continuation (e.g., indicated by specific prefixes), a predefined extension label is assigned, or the previous label is reused, depending on the configuration.

Special tokens, such as `[CLS]` and `[SEP]`, are added to the tokenized text to signify the start and end of the input sequence, respectively. Sentences are truncated or padded to a fixed length to ensure uniform input sizes, with padding applied to maintain alignment between tokens, labels, and features.

At the end for each abstract, a series of features is generated:

- **Input IDs:** Numerical identifiers representing the tokens.
- **Attention Mask:** A binary mask indicating which tokens should be attended to.
- **Segment IDs:** Markers distinguishing different segments (not used).
- **Label IDs:** Numerical representations of the labels aligned with the tokens.
- **Label Probability Distributions:** For each sentence, the distribution of labels is computed over three categories: claims, premises, and outside tokens. This captures the overall argumentative structure of the text.

The preprocessing steps involving reading input data in a CoNLL-style format and generating Input IDs, Attention Masks, Segment IDs, and Label IDs were adapted from (Mayer et al., 2021). In contrast, the computation of the Label Probability Distribution was developed as part of this work.

6 Experimental setup and results

This section outlines the experimental procedures and settings employed to evaluate the proposed models for argumentative quantification in RCT abstracts.

6.1 Training and Evaluation Details

All models were trained using a unified framework implemented in PyTorch. Each model was trained for a maximum of 20 epochs. A batch size of 1 was adopted for the training set and 32 for the evaluation set. The optimizer used was AdamW, configured with a learning rate of 2×10^{-5}, weight decay of 0.1, and epsilon set to 1×10^{-8}. Additionally, trainable parameters were divided into groups with and without weight decay, ensuring biases and layer normalization weights were excluded from weight decay.

A linear learning rate scheduler with 4 warmup steps was employed to gradually increase the learning rate during the initial steps before linearly decaying it throughout the rest of the training. Gradient clipping was applied, with gradients' norm capped at 3.0 to enhance stability. The training datasets comprised input IDs, attention masks, token type IDs, true labels, and, where relevant, probability distributions over labels. For tokenization, I utilized the pre-trained `SciBERT` model, that is trained on scientific data, (Beltagy et al., 2019), specifically `allenaiscibert_scivocab_uncased`, configured to lowercase the input text.

The evaluation process was conducted at the end of each epoch using a separate validation dataset.

Evaluation involved computing the loss and several metrics, including KLDivergence, MAE and MSE, applied to both the probability distribution and to the claims.

6.2 Results

The training histories, illustrated in Figure 1, reveal notable differences in metric values across the models. From the plot, it is evident that the **classify and count** model achieved the lowest values across all metrics, indicating superior performance. On the other hand, the **quantify model using MAE loss** exhibited significantly worse validation metric values, particularly highlighting its limitations in comparison.

To further substantiate these findings, Table 1 presents the metrics computed on the test set for all three models.

Table 1: Test Set Metrics for Different Models

Metric	Classify and Count	BERT Quantify KLDiv	BERT Quantify MAE
KLDivergence	0.0296	0.0445	0.0962
MeanAbsoluteError	0.0462	0.0614	0.0684
MeanSquaredError	0.0056	0.0066	0.0083
MAEForClaims	0.0396	0.0471	0.0547
MSEForClaims	0.0034	0.0039	0.0054

7 Discussion

The results from the test set metrics clearly indicate that the **classify and count** model consistently outperforms the others. In particular, it achieves better accuracy across all metrics, while the **BERT quantify using MAE loss** model exhibits notably higher errors. This suggests that not only does the **classify and count** model excel in predicting the distribution probabilities of content within an abstract (such as claims, premises, and outside), but it also demonstrates strong performance in predicting the quantity of claims within an abstract—an essential application for which these models were specifically developed.

Interestingly, the approach based on direct probability distribution prediction is less accurate than the **classify and count** method, which employs POS tagging and aggregates its results. This finding reveals that a model can perform better on one task (predicting the amount of claims) by solving a related but different task (POS tagging) rather than directly addressing the original task.

8 Conclusion

In conclusion, this paper has explored two distinct approaches for the automatic quantification of argumentative components in abstracts from Randomized Controlled Trials (RCTs). The first approach, the **classify and count** model, leverages a sequence classification framework using POS tagging, followed by the aggregation of token-level predictions to estimate the probability distribution of claims, premises, and outside components. The second approach directly estimates the distribution by predicting the proportions of claims, premises, and outside components.

My results demonstrate that the **classify and count** model outperforms the direct quantification approach, particularly in accurately estimating the proportion of claims within the abstracts. This underscores the efficacy of combining POS tagging with aggregation techniques for the task at hand, and highlights the advantages of a more structured, indirect approach over direct distribution prediction.

For future work, an interesting direction would be to explore a hybrid approach, in which two separate heads could be used, one for POS tagging and another for directly predicting the probability distribution. A weighted average of the losses from both heads could be used to optimize the model. Additionally, experimenting with different BERT models, beyond SciBERT, could provide valuable insights into whether changes in model architecture lead to performance improvements in this task.

9 Links to external resources

The link to the GitHub repository is the following: https://github.com/DanieleBaiocco/NLPProjectWork.

This is, instead, the link to the dataset I used: https://gitlab.com/tomaye/abstrct.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: Pretrained language model for scientific text. In *EMNLP*.

Tobias Mayer, Elena Cabrio, and Serena Villata. 2020. Transformer-based argument mining for healthcare applications. In *ECAI 2020 - 24th European Conference on Artificial Intelligence*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 2108–2115. IOS Press.

Tobias Mayer, Santiago Marro, Elena Cabrio, and Serena Villata. 2021. Enhancing evidence-based

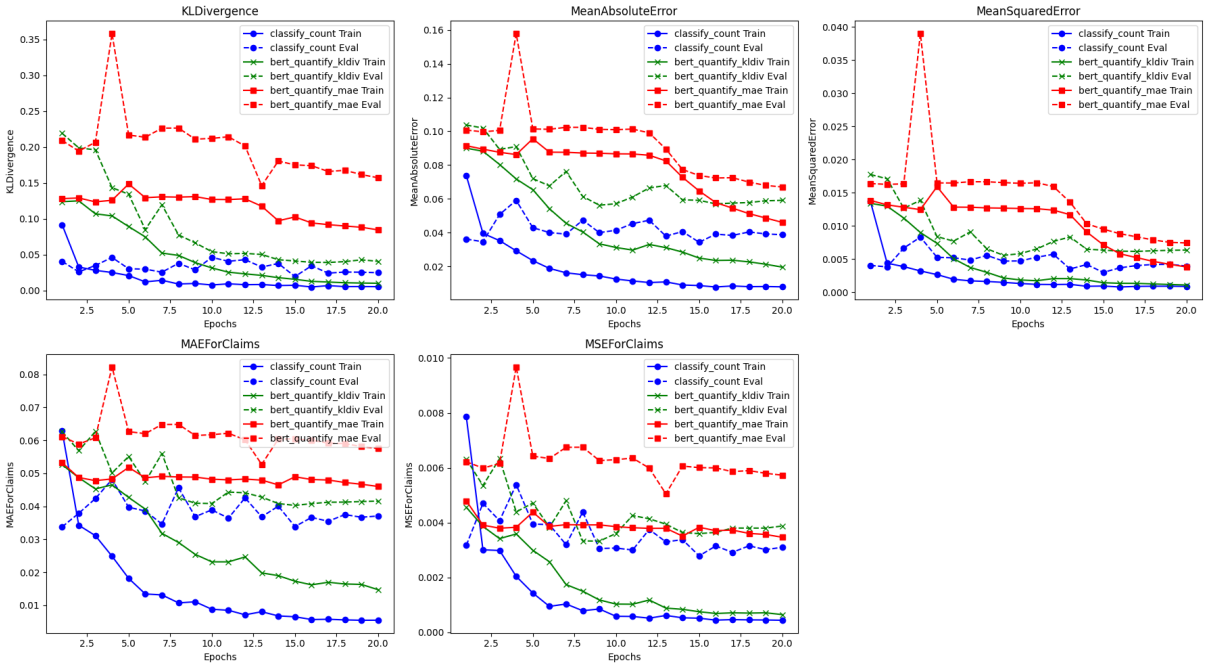

Figure 1: Metric values across training histories for all models.

medicine with natural language argumentative analysis of clinical trials. *Artificial Intelligence in Medicine*, 118:102098.

Alejandro Moreo, Andrea Esuli, and Fabrizio Sebastiani. 2021. Quapy: A python-based framework for quantification. *CoRR*, abs/2106.11057.