Opinion Mining Using Pre-Trained Large Language Models: Identifying the Type, Polarity, Intensity, Expression, and Source of Private States

Saeed Ahmadnia[⋆†], **Arash Yousefi Jordehi**[⋆†], **Mahsa Hosseini Khasheh Heyran**[⋆†], **Seyed Abolghasem Mirroshandel**[⋆†] **and Owen Rambow**[♣◇‡]
[⋆] Department of Computer Engineering [♣] Department of Linguistics
[◇] Institute for Advanced Computational Science
[‡] Stony Brook University, Stony Brook, NY, USA. [†] University of Guilan, Rasht, Guilan, Iran.
saeed.ahmadnia@outlook.com, arashy76@phd.guilan.ac.ir, mahsahsii@gmail.com,
mirroshandel@guilan.ac.ir, owen.rambow@stonybrook.edu

Abstract

Opinion mining is an important task in natural language processing. The MPQA Opinion Corpus is a fine-grained and comprehensive dataset of private states (i.e., the condition of a source who has an attitude which may be directed toward a target) based on context. Although this dataset was released years ago, because of its complex definition of annotations and hard-to-read data format, almost all existing research works have only focused on a small subset of the dataset. In this paper, we present a comprehensive study of the entire MPQA 2.0 dataset. In order to achieve this goal, we first provide a clean version of MPQA 2.0 in a more interpretable format. Then, we propose two novel approaches for opinion mining, establishing new high baselines for future work. We use two pre-trained large language models, BERT and T5, to automatically identify the type, polarity, and intensity of private states expressed in phrases, and we use T5 to detect opinion expressions and their agents (i.e., sources).

Keywords: Corpus, Opinion Mining/Sentiment Analysis, Statistical and Machine Learning Methods, Large Language Models, MPQA

1. Introduction

Sentiment analysis, also called opinion mining, is the research area that investigates how people express opinions, sentiments, and attitudes towards targets including events, individuals, products, services and their attributes (Liu, 2020). By analyzing a text, we can identify the subjective information in context. A subjective expression is "any word or phrase used to express an opinion, emotion, evaluation, stance, speculation, etc." (Wilson et al., 2005). A general covering term for such states is "private state" (Quirk et al., 1985).

In the past few years, Deep Learning (DL) techniques have been broadly used because of the increase in available computational power and because of the vast amount of available data on the Internet. NLP is one of the fields in which DL has had a huge influence (Karimi et al., 2021).

BERT (Devlin et al., 2019) and T5 (Raffel et al., 2022), are two pre-trained deep language models which are used in many NLP tasks and show successful results (Gao et al., 2019; Sun et al., 2019; Hoang et al., 2019; Zhang et al., 2021; Raval et al., 2021; Xue et al., 2021). BERT creates a deep understanding of language in textual data by utilizing surrounding text to establish context. T5 is a generative text-to-text model which can be fine-tuned on diverse tasks simultaneously. Furthermore, FLAN-T5 (Chung et al., 2022) is a variant of T5 which led to considerable improvements on a variety of tasks over the T5 large language model (Murzaku et al., 2023).

In our research, we use BERT and T5 pre-trained large language models to build a system consisting of several models for solving different opinion mining problems on the Multi-Perspective Question Answering (MPQA) fine-grained corpus. We compare the results obtained using each pre-trained model.

The contributions of this paper are as follows: 1) We release a set of tools that can be used to automatically derive and clean all types of private states available in MPQA 2.0 and provide the dataset in an easily usable format. This is the first MPQA cleaning code released to the community. 2) We present new high baselines for two groups of tasks related to MPQA: determining type, polarity, and intensity of private states, and finding opinion expressions and sources of private states. 3) We perform extensive experiments related to these baselines, comparing different model architectures, input and output formats, and learning paradigms. The MPQA cleaning code and model implementations can be found at: https://github.com/theSaeed/opinion-mining-using-llms.

The structure of the paper is as follows. We explain the background and dataset in Section 2. We discuss related work in Section 3. In Section 4, we propose some approaches to utilizing the dataset. Next, we adopt the mentioned approaches and

LREC-COLING 2024, pages 12481–12495
20-25 May, 2024.

show the experimental results and discussions in Section 5. Finally, in Section 6, we draw conclusions and state our future work.

2. Background and Dataset

The MPQA 2.0 opinion corpus[1] contains annotations of subjective and objective expressions. The annotation schemes and the details of the collection methods of this corpus are described in Wiebe et al. (2005). We briefly summarize some of the key points about MPQA which are important for our research, and then describe the revision of the MPQA corpus we have created.

2.1. The MPQA Corpus: Basics

Conceptually, a private state is the state of a **source** (the experiencer or holder of the private state, also called **agent** in the MPQA literature), holding an **attitude**, optionally toward a **target**. The annotators of MPQA were trained to annotate text at the word-level and phrase-level based on a comparatively fine-grained annotation scheme. The following **types** of attitudes were annotated in MPQA: agreement, arguing, intention, sentiment, speculation, and other-attitude (Wilson, 2008).

MPQA annotates **nested sources** rather than plain sources. Consider a sentence in which the author writes about other people's private states and speech events. Such cases lead to multiple sources in one sentence, and actually form an according-to relation among sources (Murzaku et al., 2023). However, we only learn about the other sources from the author. Therefore, we have a nesting of sources in a sentence. For instance, in the sentence ***Robert*** *said that* ***Paul*** *hates* ***Donald***, the nested source of the attitude expressed by *hate* is expressed as $[Author, Robert, Paul]$. This notion of "nested source" was also used in other annotation efforts, including FactBank (Saurí and Pustejovsky, 2009).

Private states can be characterized along two dimensions (Tian et al., 2018): **polarity** and **intensity**. Polarity is the fundamental problem in sentiment analysis (Akkaya et al., 2009): is the sentiment positive or negative? The notion extends to other types of private state such as arguing (for or against). Intensity is the degree of strength with which a source holds a private state.

A private state is signaled in text through an **opinion expression**. There are three types of opinion expressions. Subjective private states can either be expressed explicitly (Direct Subjective (DS)) or implicitly (Expressive Subjective Element (ESE)). DS expressions also include subjective speech events (e.g., *criticize*). Objective Speech Event (OSE) expressions do not convey any subjectivity. These opinion expressions are the anchors for the annotation in MPQA.

MPQA annotates private states using two types of *frames*: opinion expressions and attitudes. Each opinion expression annotation item has a *type* (DS, ESE, or OSE), a *polarity*, an *intensity*, and a *nested-source* field; not all are required in an annotation. DS expressions (but not ESE or OSE expressions) are linked to attitudes, which are annotated separately. Attitudes have a *type* (agreement, arguing, intention, sentiment, speculation, or other-attitude), a *polarity*, an *intensity*, a *nested-source*, and a *target* field.

The *nested-sources* field is an ordered list, in which the first item is the author (writer), and then followed by other sources ordered by the "according-to" relation. Polarity is *positive*, *negative*, or *neutral* for opinion expressions, and *positive* or *negative* for attitudes. There are four possible class labels for intensity: *low*, *medium*, *high*, and *extreme*. Additionally, there are some instances labeled as *low-medium*, *medium-high*, and *high-extreme* in the dataset which are not defined in the MPQA annotation scheme. These middle-type classes demonstrate the uncertainty of annotators, which makes the problem much harder especially for machines. However, we include them in our experiments. Note that, similar to Choi and Cardie (2010), the labels *extreme* and *high-extreme* were merged into *high* in our experiments due to their limited occurrence. For illustration, consider Example 1 and its corresponding annotation items in Table 1 (as annotated in MPQA).

Ex 1. *A couple of weeks ago, the US state department in its annual report had revealed that Iran, Pakistan and some other countries are the violators of human rights.*

A key point and focus of MPQA work is analyzing private state expressions in context, instead of judging words and phrases independently, out of context. As a result, this corpus is well-suited for studying ambiguities that arise in subjective language. One of these ambiguities is the word-sense ambiguity, e.g., the objective sense of *interest* as in *interest rate* for financial uses versus the subjective sense as in *take an interest in*. Another case of ambiguity is in the idiomatic versus non-idiomatic usage, e.g., the word *bombed* in Example 2 versus Example 3. Irony, sarcasm, and metaphor, which are categorized as pragmatic ambiguities, can also be added to the list of ambiguities.

Ex 2. *The famous comedian absolutely bombed last weekend.*

Ex 3. *The army bombed the automobile company.*

[1] http://mpqa.cs.pitt.edu/

Expression	Annotations			
	Type	Polarity	Intensity	Nested-Source
[IMPLICIT] (whole sentence)	OSE	-	-	[AUTHOR]
in its annual report had revealed	DS	-	medium	[AUTHOR], USSD
are the violators of human rights	ESE	negative	medium	[AUTHOR], USSD
are the violators of human rights	arguing	positive	medium	-
are the violators of human rights	sentiment	negative	high	-

Table 1: Annotations for Example 1; "USSD" = "the US state department"; the first three lines are annotations of expressions, and the last two lines are annotations of attitudes that are linked to the DS in line 2 and that inherit their source from it implicitly.

2.2. Our Revision of MPQA

We found several errors related to offsets when we examined the original dataset: some annotation items that were marked with an incorrect offset, and some documents in which all annotations' offsets were wrong. We fixed some of these annotations and removed others. Then, we transformed the original MPQA annotation format to a JSON format. Additionally, we have added two more attributes to each annotation item that contain cleaner versions of the sentence and the opinion expression by removing special characters and HTML and XML tags that were present in the annotated text. Note that we did not use XML as the base format for converting MPQA to JSON. Instead, we started with the MPQA format, which is a general stand-off annotation type. The mention of HTML and XML tags is because of their presence in some documents within the MPQA corpus. Overall, this cleaning process makes the dataset easier to read and interpret.

Some of the specific modifications we made to the dataset are as follows:

- We addressed incorrect spans in 175 annotation items across 21 documents by rectifying the spans. (e.g., 'the repor' → 'the report')
- In one document, 58 annotation items contained typographical errors, which we corrected. (e.g., 'InhumanelyAs' → 'Inhumanely As')
- We noticed that 5 agent annotation items in 5 documents had apostrophes within their spans. We removed these apostrophes from their spans. (e.g., "President Bush'" → "President Bush")
- In 2 documents, we found 133 annotation items that were incorrectly annotated, and they could not be matched with any of the phrases in those documents. Consequently, we removed them.
- Additionally, 1,526 annotation items had to be removed since they could not be matched with any of the sentences as their containers.
- We also conducted a cleaning process for 206 annotation items that contained special characters and tags.

The distribution of all annotation types (e.g., ESE, DS, OSE) available in the original and our cleaned version of MPQA is shown in Table 2. As can be seen, the number of some items decreased due to the mentioned problems in the original dataset. We share our code (which encompasses the entire cleaning process) to generate the cleaned JSON version of MPQA on this GitHub repository.

Annotation Type	Original	Our Version
Agents	14,595	14,562
Targets	8,413	8,397
ESE	13,793	13,654
Sources in ESE	21,497	21,411
DS	15,437	15,076
Sources in DS	31,253	30,567
Attitudes in DS	10,336	9,973
OSE	16,906	15,959
Sources in OSE	22,542	20,794
Attitudes	10,308	10,292
Targets in Attitudes	9,466	9,056
Sentences	15,789	15,789

Table 2: Number of annotation items with the specified types in the original and our polished version of the MPQA dataset.

3. Related Work

Sentiment analysis at the document and sentence level is not effective in accurately determining the preferences of individual entities or identifying opinions related to specific entities or topics (Liu, 2020). As discussed in Section 2, the MPQA dataset has been annotated at the word and phrase level rather than at the sentence or document level, using a complex annotation scheme (Wiebe et al., 2005). Its annotations reflect how language expresses the private states of the author and of the author's perception of the private states of the agents mentioned in the document. MPQA thus moves away

from sentiment being a property of text to sentiment being something people have about something.

Early research using the MPQA dataset primarily relied on traditional Machine Learning (ML) methods, which required extensive feature engineering and relied on linguistic resources and manually created sentiment lexicons. Contextual polarity prediction was introduced by Wilson et al. (2005), employing hand-designed features and a large lexicon of words utilizing a two-step classifier for neutral and non-neutral labels. Intensity and polarity classification on MPQA were also explored in Choi and Cardie (2010) by applying a hierarchical parameter-sharing technique using Conditional Random Fields. Wilson (2008) also utilized different traditional ML algorithms for recognizing the intensity, polarity, and attitudes of private states in MPQA. Similar methods were also applied on a newer version of MPQA (i.e., MPQA 3.0 (Deng and Wiebe, 2015)) on polarity detection (Deng and Wiebe, 2015).

Neural networks have the ability to learn meaningful representations from text data without the need for feature engineering (Pang et al., 2021; Gao et al., 2019). Employing diverse neural embeddings (i.e., pre-trained vectors) has been shown to yield substantial performance enhancements across a range of downstream NLP tasks. Schneider et al. (2020) conducted a study examining the impacts of various embedding combinations. "AdaSent", proposed by Zhao et al. (2015), generated a multi-scale hierarchical representation instead of a fixed-length representation. Other studies, such as Conneau and Kiela (2018) and Kim (2014), utilized CNN models trained on word representations for binary opinion polarity classification at the sentence level.

In other research on MPQA, semantic and syntactic structures have been used to find opinion expressions and their holders (Johansson and Moschitti, 2010b,a). Some other research mainly focuses on holders (agents) and targets of opinion expressions (Marasović and Frank, 2018; Zhang et al., 2019), and some work jointly predicts each agent and target toward the corresponding opinion expression (Xia et al., 2021) or polarity of an opinion expression concurrently (Johansson and Moschitti, 2011).

Two of the most successful previous works, Xia et al. (2021) and Wu et al. (2022), have proposed an end-to-end span-based method using BERT to extract opinion expressions, as well as their holders (i.e., sources) and targets. Zhang et al. (2020) and Katiyar and Cardie (2016) also focused on solving the same problem utilizing dependency graph convolutional networks and bidirectional LSTMs, respectively.

To the best of our knowledge, this paper is the first to predict the type of a private state; previous work presumes that the private state type is given, or they do experiments on a specific type only (Xia et al., 2021; Wu et al., 2022). Furthermore, we are also the first to predict the source(s) of a private state while explicitly preserving the nested hierarchy. Finally, to our knowledge, we are the first to predict opinion expressions in a generative manner. Even if we are tackling the same task as previous work, it is not possible to directly compare our work to the other existing research on MPQA because we use our novel complete and cleaned version of MPQA. For comparison, the MPQA dataset used in many recent research papers contains only a subset of the DS annotations that are available in our dataset (Xia et al., 2021; Zhang et al., 2020, 2019; Marasović and Frank, 2018; Katiyar and Cardie, 2016), without any ESE and OSE annotations. Furthermore, there is no related research investigating other aspects of private states, such as intensity and polarity, in the way we are doing it.

4. Our Approach

We put forward various models to address two distinct sets of problems. The initial set comprises the classification of private states' type, polarity, and intensity based on the context. Our second set of problems involves identifying the expressions and sources of private states through a generative approach. Please note that our research did not specifically address target extraction, and we plan to address this task in our future studies.

4.1. Type, Polarity, and Intensity

Our goal in this approach is to automatically identify the *type*, *polarity*, and *intensity* of private states expressed in phrases, given the phrase and the sentence it occurs in. For achieving this goal, we first build and fine-tune separate models for predicting each attribute. Then, we build a universal model that can predict all of these attributes at the same time. To get a proper comparison, we omitted the samples (about 7.7% of the dataset) in which the value of any of these attributes (i.e., *type*, *polarity*, and *intensity*) is missing. As we are interested in subjective language, we omit the OSE type, and we also omit DS annotations because they are associated with attitudes, and we include all the attitude type of the DS instead of DS as type. As a result, we end up with the following labels and their distribution for each task:

- Type: *agreement* (×284), *arguing* (×2,466), *expressive subjectivity* (×11,604), *intention* (×420), and *sentiment* (×3,862).

- Polarity: *negative* (×7,844), *neutral* (×5,583), and *positive* (×5,209).
- Intensity: *low* (×4,837), *low-medium* (×1,262), *medium* (×7,574), *medium-high* (×1,258), *high* (×3,705).

4.1.1. Solo Models

In this section, we build models for each of the type, polarity, and intensity tasks individually. Each sentence and expression can simultaneously express various private states of any type. So, for the type task, we use a sentence and a gold-standard expression as input and predict what types of private states the expression communicates based on its context. Subsequently, in the polarity and intensity tasks, we feed the model with the gold-standard type in addition to the sentence and the gold-standard expression. The type, polarity and intensity classification tasks are 5-class multi-label, 3-class single-label, and 3-class multi-label tasks, respectively. However, we transform the 3 classes for intensity into 5 labels, so for evaluation purposes, it becomes a single-label task. For example, for intensity, the label "medium-high" in encoded as "medium" and "high".

We designed a comprehensive set of models utilizing BERT and T5 as our base models in these tasks. We also tried feeding the model with different kinds of input formats (Section 5.1, Table 3). Furthermore, we used additional fully connected layers and various CNN and RNN (e.g., multi-layer GRU and bidirectional LSTM) layers in parallel, in continuation, or even without the use of the pre-trained models (i.e., training from scratch). Finally, another method that we used is to combine the Layer-wise Learning Rate Decay (LLRD) (Sun et al., 2019) with freezing the first few layers of the base model.

4.1.2. Universal Model

In addition to solo models, we provided a model that can simultaneously identify the type, polarity, and intensity of an expression in a sentence. We have 5 units in the output to represent the labels of the type task. Then, we have a pair of 3 units for each type, so that we can obtain the polarity and intensity of their corresponding type. We can use this new model in Single-Task Learning (STL) and Multi-Task Learning (MTL) approaches. Learning multiple related tasks jointly has the potential to improve the general performance of all the tasks at hand (Zhang and Yang, 2022).

As we now do not have the gold-standard type in the input, this model cannot be directly compared to our previous polarity and intensity models. Therefore, we use the gold-standard type like a gate and only select their respective polarities and intensities among all predictions during the evaluation process. In this way, we can directly compare our results with the previous models. For a better intuition, you can look at Figure 1. The universal model is very similar to the BERT-based models for type if we deactivate the output units for polarities and intensities. But it actually provides a new structure for each of the polarity and intensity solo tasks when we deactivate the other tasks' output units.

First, we use the universal model for the polarity and intensity tasks, without MTL, to see the impact of this new architecture. Then, we try to train the whole model using all tasks' data at the same time. We also investigate setting a single task as our main goal and give small coefficients to the other tasks' learning rates. Another exploration we conduct is to sequentially train the model on each task's data, and evaluate the model only on the last task it has been trained on. This way we can probe the impact each task has, when we transfer its related knowledge to be used in another task.

4.2. Opinion Expressions and Their Agents

The proposed approaches are end-to-end methods, which predict opinion expressions that contain *ESE*, *DS*, and *OSE* categories and their sources, which is more general than previous studies (Wu et al., 2022; Xia et al., 2021; Marasović and Frank, 2018), which only consider the *DS* category. We used two separate models for each of these tasks (i.e., one model for expression detection and another model for source extraction).

Our models are based on the generative T5 architecture. T5 accepts arbitrary *prefix* terms appended at the beginning of the input. It can also work without giving any prefix terms. The input of the expressions model is the sentence, and the output is all expressions found in that sentence. We use prefixes in the input (e.g., "*DS*", "*ESE*", and "*OSE*") that tell the model which type of expressions should appear in the output. Instead of feeding the model a sentence and expecting it to generate all opinion expressions at the same time, we feed the model three separate inputs for a single sentence, each prefixed with a specific term ("DS", "ESE", and "OSE"). This strategy not only mitigates the problem of overly long output sequences but also facilitates the parsing process by providing dedicated outputs for each prefix category.

About agents of expressions, we may have more than one nested-source in a text that the proposed models are able to extract. One way of feeding the source generation system is similar to the model for opinion expressions. In this setting, we give sentences with prefixes at the beginning in order

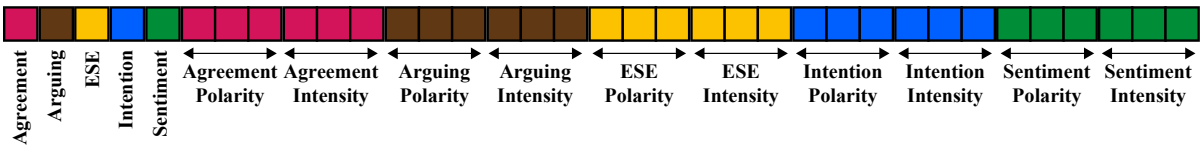

Figure 1: Universal model output units.

to generate that type of nested-sources (denoted "PS" in Tables 12 and 13). Hence, in the output, we will have a sequence of nested-sources, and each nested-source comprises one or more sources. In the second setting, we pass sentences without any prefix as the input, and at the output we will have each type's nested-sources (denoted "S" in Tables 12 and 13). Further details about input/output of these methodologies is available in Appendix 12.

5. Experiments

For all of our experiments, we report the average value of the 5-fold cross-validation results. There are many models which we trained and evaluated in each of these tasks. As mentioned in section 3, it is not possible to compare our work to prior research, due to differences in tasks, labels, and/or amount of used data. The exact configurations and hyper-parameters utilized in our experiments can be observed in Appendix 11.

5.1. Naming Scheme of Models

We employ the following naming convention for each model:

> "LM Name" : "Input Format" _ "Output Format" _ "Model Architecture"

Briefly, each scheme has four parts. The first part is the name of the pre-trained large language model used (i.e., "LM Name"). In our experiments "LM Name" can be *B* for BERT, *T5*, or *FLAN-T5*. The second part is the input formatting (i.e., "Input Format"). Then for BERT, the naming continues with output formatting (i.e., "Output Format") that is the selected part of the BERT output intended for classification. Subsequently, in the fourth part for BERT, the name continues with the model architecture name added on top of BERT or methods used on BERT. Note that some architecture models are composed of a combination of several basic models. In the name of composed models, the symbol "->" is used to indicate that two basic models are connected serially.

Table 3 shows the types of "input format". The types of "Output Format" of the BERT embeddings are shown in Table 4. And the different types of basic models architecture are as Table 5.

As an example, in the sequence *B:TES_CLS_1FC*, the *B* shows that this setting is based on BERT. At first, the input will be passed as *TES* formatting to BERT. Then, the corresponding output for the *[CLS]* token of BERT will be forwarded to a fully connected layer.

5.2. Results of Type, Polarity, and Intensity

In our experiments, we start with BERT-base-uncased as our base model. During the first set of experiments, we feed the model with different input formats and continue with the one that achieved the best result. Because of the greedy nature of the choice of experiments, we do not always test the same architectures on all tasks.

Second, we use the universal model (Section 4.1.2). Since the architecture of the type task is almost the same as the universal model when we only consider the type output units, we do not use the universal model for type. Specifically, if we deactivate the polarity and intensity output units in the universal model, there will only remain five active units at the end which is similar to the previous type task's architecture. Hence, we do not specify whether we have used the solo or the universal model in our type experiments. We use both STL and MTL approaches in our universal models.

Third, we try different variations of T5 with various input formats and get our overall best results on the type, polarity, and intensity tasks by using FLAN-T5-Base as our base model. We report F1 score (micro-average for the type task, and weighted-average for the polarity and intensity tasks) in these experiments. The STL results are shown in Tables 6 (type), 7 (polarity), and 8 (intensity). The maximum result we can get by choosing the majority class is denoted in the tables as the majority baseline. Additionally, the accuracy and F1 score results for two different MTL approaches can be found in Tables 9 and 10. The most successful settings are indicated in bold in these tables.

Name	Description
TESC	*"[CLS]" + Type + "[SEP]" + "[CLS]" + Expression + "[SEP]" + "[CLS]" + Sentence + "[SEP]"*.
TES	*"[CLS]" + Type + "[SEP]" + Expression + "[SEP]" + Sentence + "[SEP]"*.
TSE	*"[CLS]" + Type + "[SEP]" + Sentence + "[SEP]" + Expression + "[SEP]"*.
SCTTID	*"[CLS]" + Type + "[SEP]" + Sentence + "[SEP]"* is given as input; but the elegant point is that we change the *expression* segment of the *sentence* token type IDs in order to distinguish between the *expression* and *sentence*.
sepSET	*Sentence*, *Expression*, and *Type* are given to the model separately, and then the three obtained outputs are merged together.
SpE	*Sentence + "|" + Expression + "</s>"*.
EpS	*Expression + "|" + Sentence + "</s>"*.
indicatorSpE	*"sentence=" + Sentence + "|" + "aspect=" + Expression + "</s>"*.
indicatorEpS	*"aspect=" + Expression + "|" + "sentence=" + Sentence + "</s>"*.
TpSpE	*Type + "|" + Sentence + "|" + Expression + "</s>"*.
TpEpS	*Type + "|" + Expression + "|" + Sentence + "</s>"*.
TcEpS	*Type + ":" + Expression + "|" + Sentence + "</s>"*.
indicatorEpSpT	*"aspect=" + Expression + "|" + "sentence=" + Sentence + "|" + "type=" + Type + "</s>"*.

Table 3: Naming scheme for the input format of language models. The + symbol means concatenation of strings. We denote BERT classification token and separation token as *[CLS]* and *[SEP]*, respectively. "</s>" indicates a special token used in the T5 language model.

Name	Description
CLS	Only take the *[CLS]* token embedding (first token of BERT embedding).
ALL	Utilize all of the token embeddings.
EXP	Select the token embedding(s) within the *expression* span.
NLayers	Use the last N layers of BERT.
NCLS	Use the N *[CLS]* token embedding (N is an integer that is $N \geq 2$). Note that for this kind, the type of input formatting must have at least N *[CLS]* tokens.

Table 4: Naming scheme of output format of BERT embedding.

Name	Description
NFC	N-layer Fully-connected network (N is an integer that is $N \geq 1$).
BLSTM	Bidirectional LSTM.
BGRU	Bidirectional GRU.
MCNN	Multi-channel CNN (Kim, 2014).
CAT	Concatenate the output of the previous part.
MAX/MIN/SUM	Calculate the maximum, minimum, or sum of the output of the previous part.
LWS	Use learnable weighted sum.
LLRD&F	Use the LLRD method in addition to freezing the first layers of the base model.

Table 5: Naming scheme of basic models architectures.

Model Name	F1
Majority Baseline	63.7
B:ES_CLS_1FC	82.8
B:ES_ALL_BGRU->1FC	82.9
B:ES_CLS_LLRD&F_1FC	83.5
B:ES_CLS_1FC (MTL)	82.4
T5:SpE	83.1
FLAN-T5:SpE	83.6
FLAN-T5:EpS	83.2
FLAN-T5:indicatorSpE	83.5
FLAN-T5:indicatorEpS	**83.8**

Table 6: Evaluation results of type classification.

Model Name	F1
Majority Baseline	24.9
B:TESC_CLS_1FC	85.5
B:TESC_CLS_5FC	85.5
B:TESC_3CLS_3FC->CAT->1FC	85.5
B:sepSET_CLS_LWS->SUM->1FC	83.2
B:TESC_ALL_BGRU->4FC	85.2
B:TESC_ALL_BGRU->MCNN->1FC	85.4
B:TESC_ALL_MCNN->1FC	85.4
B:TESC_ALL_MAX(BGRU->4FC,MCNN->1FC)	85.6
B:TESC_4Layers_CAT->1FC	85.6
B:TESC_12Layers_LWS->SUM->1FC	85.7
B:ES_CLS_1FC (STL Universal)	85.8
B:ES_CLS_LLRD&F_1FC (STL Universal)	86.1
B:ES_CLS_1FC (MTL Universal)	85.7
T5:TpSpE	87.0
FLAN-T5:TpSpE	86.8
FLAN-T5:TpEpS	87.0
FLAN-T5:TcEpS	86.8
FLAN-T5:indicatorEpSpT	**87.1**

Table 7: Evaluation results of polarity classification.

5.3. Discussion of Type, Polarity, Intensity Results

We can see that all of our solo BERT-based results fall in a fairly narrow band. The one exception is that the sepSET and TSE input formats perform markedly worse than the TES format.

The universal model outperforms the solo BERT-based architecture for the polarity and intensity tasks (recall that it was the same architecture for the type task). When we use the LLRD and freezing layers approaches, it further improves the results for type and polarity (while remaining constant for intensity). Most tasks' performances reduce slightly when we do MTL instead of STL. We conducted experiments involving MTL approaches and found that by considering a task as our main task and setting the learning rates of the other tasks to a lower rate, we get a similar or lower performance. Likewise, the results of sequentially training the model for type, polarity, and intensity tasks were not promising. Specifically, training the model on each task one after the other did not yield any significant improvements. Despite our efforts, only a limited number of these experiments produced promising outcomes, and we were unable to discern any meaningful relationships between these tasks.

Model Name	F1
Majority Baseline	39.5
B:TES_CLS_1FC	71.4
B:TSE_CLS_1FC	70.2
B:SCTTID_CLS_1FC	72.0
B:SCTTID_EXP_BLSTM->1FC	71.5
B:SCTTID_ALL_BLSTM->1FC	72.1
B:SCTTID_EXP_BGRU->1FC	71.8
B:SCTTID_ALL_BGRU->1FC	71.8
B:ES_CLS_1FC (STL Universal)	72.5
B:ES_CLS_LLRD&F_1FC (STL Universal)	72.4
B:ES_CLS_1FC (MTL Universal)	72.5
T5:TpEpS	72.5
FLAN-T5:TpSpE	72.2
FLAN-T5:TpEpS	72.3
FLAN-T5:TcEpS	**73.0**
FLAN-T5:indicatorEpSpT	72.5

Table 8: Evaluation results of intensity classification.

Coefficients			ACC
Type	Polarity	Intensity	
1.0	0.0	0.0	**93.2**
1.0	0.1	0.2	**93.2**
1.0	0.1	0.5	**93.2**
1.0	0.5	0.2	93.1
0.0	1.0	0.0	86.1
0.0	1.0	0.5	**86.2**
0.2	1.0	0.5	86.1
0.3	1.0	0.5	86.1
0.0	0.0	1.0	73.2
0.0	0.1	1.0	73.2
0.0	0.2	1.0	**73.3**
0.0	0.3	1.0	72.8

Table 9: Evaluation results of MTL Universal B:ES_CLS_LLRD&F_1FC model with a single main task and two supplementary tasks. The main task is denoted with a 1.0 coefficient.

Also, we see that the FLAN-T5-based models provide the best results for type, polarity, and intensity, consistently outperforming the T5-based models. It also usually improves the results when we explicitly indicate where the sentence and the expression are in the input. Another result across all three tasks is that in most cases, the *EpS* input ordering slightly outperforms the *SpE* ordering.

After analyzing the results (F1) of each type alone in the type task (third column of Table 11) on the validation set, it can be seen that the model works best (F1: 92.6%) at predicting the ESE private states. This is not surprising, since in comparison to the other types, there is far more data available for ESE type to be used by the model to be trained on. The F1 of the other tasks are less than 84.1%. Subsequently, we can understand that agreement and intention types are easier to detect than arguing and sentiment types, despite their smaller numbers.

We can do a similar analysis for the polarity (fourth column of Table 11) and intensity (fifth column of Table 11) tasks, by splitting the predictions

Tasks' Order			F1
1st	2nd	3rd	
Type	-	-	82.8
Polarity	Intensity	Type	83.2
Intensity	Polarity	Type	**83.3**
Polarity	-	-	**86.0**
Type	Intensity	Polarity	85.7
Intensity	Type	Polarity	85.9
Intensity	-	-	**72.0**
Type	Polarity	Intensity	**72.0**
Polarity	Type	Intensity	71.6

Table 10: Evaluation results of sequential training.

Type	Count	F1 (Type)	F1 (Polarity)	F1 (Intensity)
All	All	85.1	86.7	73.5
Agreement	47	77.9	97.9	78.0
Arguing	389	71.7	94.6	82.5
ESE	1,851	92.6	81.4	67.0
Intention	69	84.1	98.5	88.0
Sentiment	618	71.2	95.1	85.1

Table 11: Evaluation results of the best performer type, polarity, and intensity (third, fourth, and fifth columns respectively) models on all types and then each type separately. The distribution of each label is also included. These results are on the validation set of the first fold.

based on their corresponding private state's type. The F1 of the polarity task for ESE private states (81.4%) is the worst as expected because this is the only private state type that can have neutral polarity. In contrast, the model reaches 94.6% F1 or higher for the other types. Still, the results on agreement and intention are better than the other two, just like the type task. The intensity also works much worse (F1: 67.0%) when predicting the intensity of ESE private states, while the others achieve F1 scores higher than 78.0%. In this case, intention and sentiment private states are easier than agreement and arguing.

5.4. Results of Opinion Expressions and their Agents

Experimental results of opinion expression and their agents detection are depicted in Tables 12 and 13, respectively. Similar to some related studies (Xia et al., 2021), we employ Precision, Recall, and F1 score (we only show F1) to evaluate our experimental results using the Exact match setting (i.e., *E F1*). Additionally, we utilize two auxiliary metrics known as Binary (i.e., *B F1*) and Proportional match (i.e., *P F1*). The binary and proportional metrics, commonly referred to as the overlap metric, assess the alignment between spans (i.e., opinion expression and their agents spans), encompassing both exact matches with gold-standard spans and partial matches where spans partially overlap with gold roles. To clarify, a binary match occurs when a span precisely aligns

with a gold-standard span, while a proportional match calculates the maximum ratio of overlap between a span and the corresponding gold-standard span.

In these tables, *S* and *PS* are input formatting that show *Sentence + "</s>"* and *Prefix + ":" + Sentence + "</s>"*, respectively.

Model Name	Type	E F1	B F1	P F1
T5:PS	ESE	**54.4**	**74.2**	**72.6**
	DS	**70.6**	**80.2**	**79.8**
	OSE	**80.7**	**82.9**	**82.9**
FLAN-T5:PS	ESE	51.6	70.3	68.4
	DS	65.8	75.2	74.7
	OSE	78.1	80.0	79.9

Table 12: Evaluation results of expression detection.

In Table 13, column "Comparison Method" has two values, which show how we compute the performance: 1) *LS*, meaning that only the last source in the list is considered (i.e., we ignore the nesting of the source) and 2) *ALO* which means the entire ordered list of nested sources is considered. In our experiments, we only considered non-author sources.

Model Name	Comparison Method	E F1	B F1	P F1
T5:S	LS	73.9	81.5	81.3
	ALO	71.0	83.7	83.4
T5:PS	LS	**83.5**	**86.1**	**86.1**
	ALO	**81.8**	**87.7**	**87.6**

Table 13: Evaluation results of source detection.

5.5. Discussion of Opinion Expressions and their Agents Results

Table 12 reveals that FLAN-T5 underperforms T5 in the expression prediction task. Initial experiments also indicated similar results for the source detection task. Consequently, we have refrained from extensively running FLAN-T5 for a large number of epochs for this particular task.

Experimental results from Table 13 indicate that using prefix terms in T5 remarkably affects the model's performance. This practically increases the size of data for training and it might be because of the data augmentation phenomenon in DL.

We also note that the margin between proportional and binary evaluation is small. It can be understood that there are not many mismatched tokens (words) between gold value and predicted value. In other words, either the generative system predicts well, or it does not predict close at all.

We analyzed the predicted expression items on the validation set. 71% of prediction errors do not share any similarities with the actual values. Among these errors, the system made incorrect predictions in 45% of cases, while in 26% of cases, the system failed to make any predictions at all. For the remaining 29% of errors, the predicted and actual values overlap in terms of word sets. Our proportional and binary metrics aim to illustrate this concept. The most frequent words causing discrepancies are: *the, to, is, of, a, it, in, that*. This list points to inconsistencies in the MPQA annotation related to function words, and as a result, it seems by removing stop-words or applying normalization, the results can be improved further without compromising the quality of the dataset.

When examining the outputs of the source generation task, it becomes apparent that in nearly 94% of prediction errors, there is no similarity between the predicted sources and the actual values. However, in cases where there is an intersection between the gold and the predicted values, the most common words causing differences are: *the, of, for, he, I*.

In the "ALO" settings, approximately 17% of items differ in the length of the nested-source list among all available nested sources. To delve into the details, in 54% of cases, the system's predicted nested sources are longer than the gold nested sources, while in 46% of cases, the opposite is true. Among the well-predicted nested sources in terms of their length, over 99% of mismatches occur in the last source item, with the remaining occurring in the penultimate source item.

6. Conclusion and Future Work

In this paper, we have focused on opinion mining using pre-trained large language models on MPQA 2.0. This corpus contains annotated data in context which enables us to discover more fine-grained opinions. After cleaning the dataset, we tackled two groups of tasks. First, we have proposed models based on BERT, T5, and FLAN-T5 in order to classify the type, polarity, and intensity of private states. Second, we used the T5 model to generate spans of agents and expressions in a sentence. In addition, we propose novel multi-task approaches to solve these problems simultaneously. Our application of large language models, particularly the generative approach, to this specific problem and dataset represents a novel contribution to the field. To the best of our knowledge, there are limited instances of utilizing large language models or generative approaches in this context, which sets our work apart from existing research. More research can still be done on the MPQA dataset, and this research is strongly simplified due to the public availability of our code to generate the cleaned and easy-to-read revision of MPQA. Applying techniques like adversarial attacks, and data augmentation can be two possible directions for our future work.

7. Acknowledgements

Mirroshandel participated in this research while a visitor to the Institute for Advanced Computational Science at Stony Brook University, and gratefully acknowledges its support. We thank anonymous reviewers for useful and constructive feedback on the submission. Mirroshandel and Rambow thanks Stony Brook Research Computing and Cyberinfrastructure and the Institute for Advanced Computational Science at Stony Brook University for access to the high-performance SeaWulf computing system, which was made possible by $1.85M in grants from the National Science Foundation (awards 1531492 and 2215987) and matching funds from the the Empire State Development's Division of Science, Technology and Innovation (NYSTAR) program (contract C210148). Mirroshandel and Rambow also thank the Institute for AI-Driven Discovery and Innovation at Stony Brook for access to the computing resources needed for this work. These resources were made possible by NSF grant No. 1919752 (Major Research Infrastructure program).

8. Limitations

In this paper, we have focused on English, and among different existing datasets, our research is on the MPQA dataset. For further studies, we can work on different languages and datasets. However, it is worth mentioning that due to the complexity and depth of semantics that we are working on, we have to start from an acceptable point (i.e., MPQA dataset), which is a complex dataset that contains several different types of annotations.

9. Ethics Statement

The current work is a fundamental research in order to learn natural languages better. Opinion mining has lots of different applications such as social media analysis, customer experience analysis, market research, and political view analysis. However, several existing models, like ours, achieved acceptable results in this field of study, we recommend to use these systems as a help for human decision making process (the results should be confirmed by a human), because the fairness and robustness of these models are still an unsolved issue, and the work of human experts cannot be replaced by the output of these systems. In addition, we do not predict any ethical concerns from the algorithms and technologies proposed in this work. We have utilized publicly available dataset, language models, and open source libraries.

10. Bibliographical References

Cem Akkaya, Janyce Wiebe, and Rada Mihalcea. 2009. Subjectivity word sense disambiguation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 190–199, Singapore. Association for Computational Linguistics.

Yejin Choi and Claire Cardie. 2010. Hierarchical sequential learning for extracting opinions and their attributes. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 269–274, Uppsala, Sweden. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Lingjia Deng and Janyce Wiebe. 2015. MPQA 3.0: An entity/event-level sentiment corpus. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1323–1328, Denver, Colorado. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Zhengjie Gao, Ao Feng, Xinyu Song, and Xi Wu. 2019. Target-dependent sentiment classification with bert. *IEEE Access*, 7:154290–154299.

Mickel Hoang, Oskar Alija Bihorac, and Jacobo Rouces. 2019. Aspect-based sentiment analysis using BERT. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 187–196, Turku, Finland. Linköping University Electronic Press.

Richard Johansson and Alessandro Moschitti. 2010a. Reranking models in fine-grained opinion analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 519–527, Beijing, China. Coling 2010 Organizing Committee.

Richard Johansson and Alessandro Moschitti. 2010b. Syntactic and semantic structure for opinion expression detection. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 67–76, Uppsala, Sweden. Association for Computational Linguistics.

Richard Johansson and Alessandro Moschitti. 2011. Extracting opinion expressions and their polarities – exploration of pipelines and joint models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 101–106, Portland, Oregon, USA. Association for Computational Linguistics.

Akbar Karimi, Leonardo Rossi, and Andrea Prati. 2021. Improving BERT performance for aspect-based sentiment analysis. In *Proceedings of the Fourth International Conference on Natural Language and Speech Processing (ICNLSP 2021)*, pages 39–46, Trento, Italy. Association for Computational Linguistics.

Arzoo Katiyar and Claire Cardie. 2016. Investigating LSTMs for joint extraction of opinion entities and relations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 919–929, Berlin, Germany. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Bing Liu. 2020. *Sentiment Analysis Mining Opinions, Sentiments, and Emotions*. Studies in Natural Language Processing. Cambridge University Press.

Ana Marasović and Anette Frank. 2018. SRL4ORL: Improving opinion role labeling using multi-task learning with semantic role labeling. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 583–594, New Orleans, Louisiana. Association for Computational Linguistics.

John Murzaku, Tyler Osborne, Amittai Aviram, and Owen Rambow. 2023. Towards generative event factuality prediction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 701–715, Toronto, Canada. Association for Computational Linguistics.

Guangyao Pang, Keda Lu, Xiaoying Zhu, Jie He, Zhiyi Mo, Zizhen Peng, and Baoxing Pu. 2021. Aspect-level sentiment analysis approach via bert and aspect feature location model. *Wirel. Commun. Mob. Comput.*, 2021:5534615:1–5534615:13.

Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A Comprehensive Grammar of the English Language*. Longman, New York.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2022. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Shivam Raval, Hooman Sedghamiz, Enrico Santus, Tuka Alhanai, Mohammad Ghassemi, and Emmanuele Chersoni. 2021. Exploring a unified Sequence-To-Sequence Transformer for medical product safety monitoring in social media. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3534–3546, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Roser Saurí and James Pustejovsky. 2009. Factbank: a corpus annotated with event factuality. *Language resources and evaluation*, 43:227–268.

Rudolf Schneider, Tom Oberhauser, Paul Grundmann, Felix Alexander Gers, Alexander Loeser, and Steffen Staab. 2020. Is language modeling enough? evaluating effective embedding combinations. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4739–4748, Marseille, France. European Language Resources Association.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *Chinese Computational Linguistics*, pages 194–206, Cham. Springer International Publishing.

Leimin Tian, Catherine Lai, and Johanna Moore. 2018. Polarity and intensity: the two aspects of sentiment analysis. In *Proceedings of Grand Challenge and Workshop on Human Multimodal*

Language (Challenge-HML), pages 40–47, Melbourne, Australia. Association for Computational Linguistics.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2):165–210.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 347–354, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Theresa Ann Wilson. 2008. *Fine-grained subjectivity and sentiment analysis: recognizing the intensity, polarity, and attitudes of private states*. University of Pittsburgh.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Shengqiong Wu, Hao Fei, Fei Li, Meishan Zhang, Yijiang Liu, Chong Teng, and Donghong Ji. 2022. Mastering the explicit opinion-role interaction: Syntax-aided neural transition system for unified opinion role labeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11513–11521.

Qingrong Xia, Bo Zhang, Rui Wang, Zhenghua Li, Yue Zhang, Fei Huang, Luo Si, and Min Zhang. 2021. A unified span-based approach for opinion mining with syntactic constituents. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1795–1804, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Bo Zhang, Yue Zhang, Rui Wang, Zhenghua Li, and Min Zhang. 2020. Syntax-aware opinion role labeling with dependency graph convolutional networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3249–3258, Online. Association for Computational Linguistics.

Meishan Zhang, Peili Liang, and Guohong Fu. 2019. Enhancing opinion role labeling with semantic-aware word representations from semantic role labeling. 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 641–646, Minneapolis, Minnesota. Association for Computational Linguistics.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2021. Towards generative aspect-based sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 504–510, Online. Association for Computational Linguistics.

Yu Zhang and Qiang Yang. 2022. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609.

Han Zhao, Zhengdong Lu, and Pascal Poupart. 2015. Self-adaptive hierarchical sentence model. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, page 4069–4076. AAAI Press.

11. Implementation Details

We implemented our models with PyTorch[2] and used the Natural Language Toolkit package[3], scikit-learn library[4], NumPy[5], and Matplotlib[6]. We adopted the BERT and T5 models and their tokenizers from the *Hugging Face* Transformers library[7] (Wolf et al., 2020).

In general, our models can be stated as two categories: 1) BERT-based models. 2) T5-based

[2]https://pytorch.org/
[3]https://www.nltk.org/
[4]https://scikit-learn.org/stable/
[5]https://numpy.org/
[6]https://matplotlib.org/
[7]https://github.com/huggingface/transformers

models. In the first category, our models usually have at least nearly 109.5M trainable parameters. In the second category, our models usually have at least 222.9M trainable parameters. In the case of the FLAN-T5 version of T5, it has more than 247.5M trainable parameters.

We set the random seed to a fixed number to make the results reproducible. We used a mixture of early stopping on validation data and manually tuning hyper-parameters to find the good ones. For instance, in the BERT-based type, polarity, and intensity prediction tasks, we found the models overfitting after 3 or 4 epochs. It could be due to the fact that we did not turn off most of the trainable parameters. In type prediction and intensity prediction tasks, we adopted the binary cross entropy loss function. However, in the polarity classification problem, the cross entropy loss for multi-class classification is employed.

In each iteration, we train our models on the train set and tune the hyper-parameters according to the validation set. At last, an evaluation would be done on the test set in order to give us a fair judgment. Of course, we cannot access the test set during the training and hyper-parameter setting and it is just only available after this process. With such an approach, we can assure that the results are measured in a suitable way. The best hyper-parameter values of the BERT-based solo models, BERT-based universal models, and T5-based solo model experiments are given in Tables 14, 15, and 16, respectively. We conducted multiple experiments with different values. Also Table 17 depicts settings of the source and expression generation tasks.

We did not use any human annotators for our research. We utilized the MPQA opinion corpus which was published by a group of researchers and annotators. In this paper, we explored among the data of MPQA corpus and cleaned them as we have discussed comprehensively in Section 2.

Task	Parameter	Value
Type	Batch size	16
	Learning rate	2e-5
	Decay Factor	1.0
	Frozen Layers	3
	Dropout rate	0.1
	Max epoch	20
	Optimizer	AdamW
	Early Stopping	1200 steps
Polarity	Batch size	16
	Learning rate	7e-6
	Weight decay	6e-6
	Dropout rate	0.1
	Max epoch	5
	Optimizer	Adam
Intensity	Batch size	16
	Learning rate	1e-5
	Dropout rate	0.1
	Max epoch	4
	Optimizer	Adam

Table 14: Hyper-parameters of BERT-based solo models.

Task	Parameter	Value
Polarity	Batch size	16
	Learning rate	2e-5
	Decay Factor	0.9
	Frozen Layers	1
	Dropout rate	0.1
	Max epoch	20
	Optimizer	AdamW
	Early Stopping	1200 steps
Intensity	Batch size	16
	Learning rate	2.5e-5
	Dropout rate	0.1
	Max epoch	20
	Optimizer	AdamW
	Early Stopping	1200 steps

Table 15: Hyper-parameters of BERT-based universal models.

12. Input/Output of Source/Expression Generation Tasks

In expression generation task, we input the model sentences appended by prefixes at the beginning. Therefore, we are determining which type of opinion expression we want to be produced in the output. In the output, opinion expressions are divided by "|" (pipe) symbol. Also, if there is one expression it would be only one expression in the output.

One way of feeding the source generation system is similar to the model for opinion expressions. In this setting we give sentences with prefixes at the beginning in order to generate that type nested-sources. Hence, at the output, we will have a bunch of nested-sources separating by "||", and each nested-source, which comprises of one or more sources, arrange source items divided by "|" symbol. In another setting, we give sentence at the input, and at the output we will have each type's nested-sources splitted by "|||" symbol. Briefly, we could formulate each kind of model's input/output as the following list.

- Prefix-based Expression prediction
  - **Input:** `prefix: sentence` (prefix could be DS, ESE or OSE)
  - **Output:** `expression 1 | expression 2 | expression 3 | ...`
- Prefix-based Source prediction

Parameter	Value
Batch size	16
Learning rate	1e-4
Dropout rate	0.1
Max epoch	20
Optimizer	AdamW
Early Stopping	1200 steps

Table 16: Hyper-parameters of T5-based solo models.

Parameter	Value
Batch size	16
Learning rate	1e-4
Dropout rate	0.1
Max epoch	120
Optimizer	Adam
Early Stopping	-

Table 17: Hyper-parameters of T5-based Source/Expression generative models.

  - **Input:** `prefix: sentence`
    (prefix could be DS, ESE or OSE)
  - **Output:** `nested source 1 || nested source 2 || ...`
    , where each nested source is as `source 1 | source 2 | ...`

- Without prefix Source prediction
  - **Input:** `sentence`
  - **Output:** `ESE nested sources ||| DS nested sources ||| OSE nested sources`
    , where each type's nested source is as `nested source 1 || nested source 2 || ...`
    , where each nested source is as `source 1 | source 2 | ...`

13. Dataset Details

Let us take a look at the annotation items in the cleaned MPQA in json format. With this new data format that we have proposed, we have access to the original links that are contained in the original MPQA dataset. Since the IDs that are used in the original dataset are not exactly unique, we have added some pre-fixes and post-fixes to the IDs so that we can make them unique. The document ID that the data is drawn from is available in this format too. It allows other researchers to only focus on some specific types of documents. In the original dataset, the highlighted expressions are indicated using document-based offsets. So, users cannot directly use the dataset in sentence-level experiments. In contrast, our dataset provides users with sentences and expressions both in text format and offset format. Additionally, the type and polarity of private states are extracted from the original attitude-type field and are divided into two fields for easier access. Two examples of the annotation items are depicted in Figures 2 and 3. In our dataset, each expression and sentence is stored as "head" and "text", respectively.

```
{
    "unique_id": "20011130/12.36.18-4189&&direct-subjective-140",
    "doc_id": "20011130/12.36.18-4189",
    "sentence_id": "20011130/12.36.18-4189&&sentence-42",
    "text": "Dr. Ren was critical of this irresponsible decision by the US Government.",
    "head_start": 43,
    "head_end": 51,
    "head": "decision",
    "belief": null,
    "polarity": "neutral",
    "intensity": "low",
    "annotation_type": "direct_subjective",
    "target_link": [],
    "attitude_link": [],
    "nested_source_link": [
        "20011130/12.36.18-4189&&agent-w",
        "20011130/12.36.18-4189&&agent-Ren",
        "20011130/12.36.18-4189&&agent-US"
    ],
    "expression_intensity": "low",
    "implicit": null,
    "w_head_span": [7, 8],
    "w_text": ["Dr.", "Ren", "was", "critical", "of", "this", "irresponsible", "decision", "by", "the", "US", "Government", "."],
    "w_head": ["decision"],
    "clean_text": "Dr. Ren was critical of this irresponsible decision by the US Government.",
    "clean_head": "decision",
    "target": [],
    "nested_source": [{"w_head_span": [0, 0], "w_head": [], "clean_head": ""}, {"w_head_span": [0, 2], "w_head": ["Dr.", "Ren"], "clean_head": "Dr. Ren"},
                      {"w_head_span": [10, 12], "w_head": ["US", "Government"], "clean_head": "US Government"}
                     ],
    "attitude": []
}
```


Figure 2: First example of annotation items.


```
{
    "unique_id": "non_fbis/15.26.56-25086&&attitude-a30",
    "doc_id": "non_fbis/15.26.56-25086",
    "sentence_id": "non_fbis/15.26.56-25086&&sentence-2",
    "text": "Brazil and Germany also call on \nother countries to ratify the protocol and work to enforce it.",
    "head_start": 24,
    "head_end": 31,
    "head": "call on",
    "belief": null,
    "polarity": "positive",
    "intensity": "low-medium",
    "annotation_type": "sentiment",
    "target_link": [
        "non_fbis/15.26.56-25086&&target-t30"
    ],
    "attitude_link": [],
    "nested_source_link": [],
    "expression_intensity": null,
    "implicit": null,
    "w_head_span": [4, 6],
    "w_text": ["Brazil", "and", "Germany", "also", "call", "on", "other", "countries", "to", "ratify", "the", "protocol", "and", "work", "to", "enforce", "it", "."],
    "w_head": ["call", "on"],
    "clean_text": "Brazil and Germany also call on other countries to ratify the protocol and work to enforce it.",
    "clean_head": "call on",
    "target": [
                {
                "w_head_span": [6, 17], "w_head": ["other", "countries", "to", "ratify", "the", "protocol", "and", "work", "to", "enforce", "it"],
                "clean_head": "other countries to ratify the protocol and work to enforce it"
                }
               ],
    "nested_source": [],
    "attitude": []
}
```


Figure 3: Second example of annotation items.