Capstone Project: Multiword Expression Extraction

Brynna Kilcline
Brandeis University
May 2024

1 Introduction

The goal of this project is to train a model on a custom dataset to extract multiword expressions using encoder and decoder large language models. Multiword expressions (MWEs) are phrases that behave as a single semantic or syntactic unit. Successfully identifying them can improve the performance of downstream tasks such as parsing and machine translation (Constant et al., 2017).

2 Data

The data come from the free sample of the Corpus of Contemporary American English (Davies, 2008-). They were split using the SpaCy sentence segmentizer into chunks of around 10 sentences (from here on out referred to as paragraphs), each annotated for multiword expressions by at least two graduate students with linguistic training. The types of multiword expressions in the dataset are idioms, verb-particle constructions, noun-noun compounds, light verb constructions, and other. For the purposes of this project, I merged the idiom and other categories, resulting in the counts displayed in Table 1. The inter-annotator agreement, measured with token-level Fleiss's Kappa, was 0.5795. The annotations later underwent adjudication (Kilcline et al., 2023).

The training data contain 303 paragraphs (80 percent), and the development and test data each comprise 38 paragraphs (10 percent).

MWE Type	Number
Noun-Noun Compound	749
Verb-Particle Construction	288
Light Verb Construction	117
Idiom	329
All MWEs	1483

Table 1: Number of instances of each type of MWE identified in the data

3 Models

3.1 GPT 3.5

I used few-shot learning on GPT 3.5 Turbo with chain-of-thought prompting, following Shen et al.'s (2023) named entity recognition prompt template. Thus, I chose five paragraphs from the training data and provided sample answers as a list of MWE candidates, whether they are in fact MWEs, and the reason why. I experimented with the examples I provided and ways of justifying the decisions, using a few data points to qualitatively evaluate the performance. Then, I chose the two best prompts and quantitatively evaluated them on the entire development set.

3.2 BERT

I built two models in PyTorch with uncased base BERT (Devlin et al., 2019) at the core. In order to pass the data to the models, I had to adjust the labels to match the BERT tokenizer. The first model a simple linear layer and softmax on top of the BERT output. The second feeds BERT's output into a conditional random field and uses Viterbi decoding. I tuned the learning rate, batch size, and number of epochs for each model.

3.3 Ensemble

I also trained models to predict a single type of multiword expression. For each type, I ran experiments varying the model and its hyperparameters. From there, I chose the highest-performing model and specification for each MWE type, regardless of whether it was trained on all MWEs or just one type. I combined the model predictions in order of their reliability, beginning with the most accurate MWE type and iteratively adding in non-conflicting predictions for other MWE types.

	Dev			Test		
Model	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
GPT 3.5	20.98	24.43	22.57	16.67	16.20	16.43
BERT	47.20	43.43	45.24	36.78	44.14	40.13
BERT-CRF	50.87	51.43	51.14	38.92	49.66	43.64
Ensemble	56.07	55.43	**55.75**	40.31	54.48	**46.33**

Table 2: Overall model performance on development and test sets

MWE Type	Model	Trained on	Batch Size	Epochs	Learning Rate
NN_COMP	BERT-CRF	NN_COMP	4	5	0.0001
V-P_CONSTRUCTION	BERT-CRF	All MWEs	16	10	0.0001
LIGHT_V	BERT-CRF	All MWEs	16	10	0.0001
IDIOM	BERT-CRF	All MWEs	8	10	0.0001

Table 3: Ensemble model specifications

4 Results & Discussion

The performance of the best model of each type is shown in Table 2. Both BERT models have a batch size of eight and were trained with 10 epochs. The one with a CRF has a learning rate of $1x10^{-4}$, while the other has a learning rate of $1x10^{-5}$. The hyperparameter specifications for the ensemble model are presented in Table 3. I use F1 as the main metric rather than accuracy because MWEs are very sparse. If the model were to predict no MWEs, the accuracy would be 94.45 on the development set and 94.61 on the test set. Thus, accuracy is not a useful metric for this task.

In terms of F1, GPT 3.5 performed particularly poorly. The results of the other models were more similar to one another, with the ensemble model performing best, almost 30 points above the GPT baseline. Notably, there is a large difference between the scores for the development and test sets, and most of the performance is lost in precision rather than recall. The discrepancy is most likely due to overfitting on the development set, which is easy to do when the size of the dataset is small.

The ensemble model results are broken down in Table 4. My experiments showed that training on only one type of multiword expression was beneficial for noun-noun compounds, the most frequent type of MWE in the data, but detrimental for the other, less common types. In addition to the frequency, the compositionality of the MWE types may contribute to those findings. The identification of idioms, light verbs, and verb particle constructions may benefit from one another because, unlike noun-noun compounds, they share the trait of having non-compositional meanings.

Overall, I find the relatively small fine-tuned models to outperform the enormous, generic models for the task of detecting multiword expressions from a small, custom dataset.

5 Presentation

I built a Streamlit application to showcase my project. It includes much of the information in this report, more details on MWEs, and a demonstration.

References

Mathieu Constant, Gülşen Eryiğit, Johanna Monti, Lonneke van der Plas, Carlos Ramisch, Michael Rosner, and Amalia Todirascu. 2017. Survey: Multiword expression processing: A Survey. *Computational Linguistics*, 43(4):837–892.

Mark Davies. 2008-. Corpus of Contemporary American English (COCA).

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.

Brynna Kilcline, Gabby Masini, Ruth Rosenblum, and Annika Sparrell. 2023. Multiword expressions.

Yongliang Shen, Zeqi Tan, Shuhui Wu, Wenqi Zhang, Rongsheng Zhang, Yadong Xi, Weiming Lu, and Yueting Zhuang. 2023. PromptNER: Prompt locating

	Dev			Test		
MWE Type	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
NN_COMP	74.19	78.41	76.24	50.94	80.60	62.43
V-P_CONSTRUCTION	47.22	50.00	48.57	37.14	44.83	40.63
LIGHT_V	29.41	50.00	37.04	41.18	63.64	50.00
IDIOM	22.22	13.95	17.14	13.16	13.16	13.16
Overall	56.07	55.43	55.75	40.31	54.48	46.33

Table 4: Ensemble model performance on development and test sets

and typing for named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12492–12507, Toronto, Canada. Association for Computational Linguistics.