Fine-grained Named Entity Recognition using only Coarse-grained labels

Sudhanshu Ranjan
sranjan@ucsd.edu

Gaurav Kumar
gkumar@ucsd.edu

1 Task assignment

- **Sudhanshu** - Wrote data pipelines of the framework to get keywords corresponding to the entities, entity embeddings, and keyword embeddings. Worked on formalizing the optimization function. Generated qualitative results and worked on reporting performance on different metrics. Contributed to writing the final report.

- **Gaurav** - Setup pipelines for training & evaluation of NER model. Wrote pipeline for generating fine-grained class embeddings, and keyword re-ranking. Coded the optimization of the primal problem and performed experiments. Performed extensive hyperparameter tuning of the NER classifier. Contributed to writing the final report.

2 Introduction

2.1 Motivation

Named Entity Recognition is a critical task in the field of information extraction. It takes natural language input and classifies each word into an entity, which becomes helpful in further downstream tasks. It helps generate structured data from a vast corpus of unstructured text across multiple domains. It also finds extensive usage in biomedical data for gene identification, DNA identification, and the identification of drug names and disease names. Distant supervision approaches (Wang et al., 2020) rely on generating the dataset using keyword-matching heuristics. Weak supervision-based approaches (Lison et al., 2020; Li et al., 2021a) focus on using a few initial seed keywords or rules and creating a framework around it to get a final NER tagger. However, most work has been focused on coarse-grained NER where each word gets tagged into very few classes (such as organization, person, location).

Fine-grained NER (Wang et al., 2020; Awasthy et al., 2020) is necessary to get a better understanding of the entities and should further help with the information extraction. If there is a sentence ’Joe Biden is the president of United States of America’, it has 2 entities, (Joe Biden, ’person’) and (United States of America, ’location’). We should be able to classify Joe Biden as ’politician’ and United States of America as ’country.’ without having labeled training data on finer classes. The focus of our work is to get fine-grained labels for the entities, given only the entities’ coarse-grained labels.

2.2 Related work

The named entity recognition task is a well-studied problem in the NLP community for which traditional feature engineering methods have been used extensively (Nadeau and Sekine, 2007). Currently, a significant fraction of state-of-the-art results are obtained using Transfomer based models (Devlin et al. (2018), Liu et al. (2019)), which are very effective in getting the contextualized embeddings for words in the sentence. In more recent times, the problem has been solved using approaches based on deep learning (Yadav and Bethard (2019)) and transformers (Liu et al. (2019), (Devlin et al., 2018)), which led to a remarkable performance in a supervised setting. Further efforts were made to solve the problem in significantly weaker settings. Examples include zero-shot NER (Hoang et al., 2021) and weakly supervised NER. In the weakly supervised setting, we have a few seed words, rules, or entities, and then an iterative framework is created on top of it to get the final results. The iterative framework keeps adding new rules or entities in every iteration (Li et al. (2021b), Gupta and Manning (2014), Niu et al. (2003)). Another set of approaches includes generating noisy labeling functions and then creating a named entity recognition tagger (Lison et al. (2020), Safranchik et al.

(2020)). Zero-shot NER refers to the scenario when we have no human supervision for the NER task. Hoang et al. (2021) uses an external knowledge base to get a NER tagger that generalizes to unseen domains. Another approach is to formulate the problem as a Natural Language Inference task, train it on the NLI dataset, and then further train the model on a NLI dataset. The final predictions are made using this new model (Link here).

2.3 Contributions

The current setting of learning fine-grained NER using weak supervision (Li et al. (2021b), Gupta and Manning (2014), Niu et al. (2003)) has not been explored much, and to the best of our knowledge, we are the first to formulate it as a representation learning problem. It can also be noted that our problem is slightly different from other previously discussed NER settings, as they try to learn the entity span and its coarse label. On the other hand, we focus on getting the fine-grained type of the entity conditioned on the entity and its coarse-grained type. We build upon the hypothesis that the keyword responsible for coarse grained class would be more similar to the relevant fine grained class compared to other fine-grained classes. Based on this hypothesis, we propose a custom loss function. We write the primal, the dual and the KKT conditions. We solve the problem using the equations. Finally, we run experiments using the proposed solutions and report the performance.

2.4 Organization of Paper

Section 3 discusses the problem statement where we discuss our problem hypothesis, develop the primal problem, and write its dual along with KKT conditions. In Section 4, we provide a brief description of the dataset. In Section 5, we mention the approach taken for solving this problem and discuss the experiments. Further, in Section 6, we provide an in-depth analysis of our results. In the final section before references, we conclude our work and discuss the future work.

3 Statement of the problem

Given a set of sentences along with its coarse-grained named entity labels. We also have a possible set of fine-grained classes. The task is to learn fine-grained labels for all the entities.

Formally : Let $s_1, s_2, ..., s_t$ be set of t sentences. Let $s_i = (w_{i1}, y_{ci1}), (w_{i2}, y_{ci2}), ..., (w_{il}, y_{cil})$ be the set of word and coarse entity pairs for a sentence of length l, where w_{ij} denotes the j-th word and y_{cij} denotes its coarse NER label. We also have a set of p fine-grained classes, $f_1, f_2, ..., f_p$.

The task is to output $y_{fi} = y_{fi1}, y_{fi2}, ..., y_{fil}$ where y_{fi} is the corresponding fine grained labels for the sentence s_i.

3.1 Approach - Hypothesis

We use explainable AI tools such as LIME (Ribeiro et al., 2016) to help us with this task. First, we find the keywords responsible for each entity's class prediction, i.e., if w_{ij} is a named entity and has class y_{cij}, then find the keywords among the sentence s_i which is responsible for predicting w_{ij} as class y_{cij}. Let the keyword be $w_{ij'}$. The main hypothesis behind our approach is described below.

Hypothesis: Let the true fine label of word w_{ij} be y_{fij}. Then Similarity$(w_{ij'}, y_{fij}) \geq$ Similarity$(w_{ij'}, f_p) \forall p$. The concept of similarity between two words is a well-studied problem in NLP, and approaches based on cosine similarity & distance between the two embeddings have been widely used.

3.2 Primal Formulation

The hypothesis uses the similarity between the keyword and class label. However, we want to tag the word w_{ij} and not the keyword $w_{ij'}$. The entities would be present in different locations in the vector space compared to the keywords. Thus we aim to learn a mapping from the keyword to the entities.

More specifically, let $x_1, x_2, ..., x_n$ be the embedding of the words tagged with the coarse grained labels and $h_1, h_2, ..., h_n$ be the embedding for top keywords responsible for these labels in $\mathbf{R}^d$. We want to learn a mapping from $\mathbf{R}^d$ to low dimensional $\mathbf{R}^k$ using an iterative approach s.t. Sim$(x_p, h_p) \geq$ Sim$(x_p, h_q) \forall q$ and Sim$(x_p, h_p) \geq$ Sim$(x_q, h_p) \forall q$. While doing so, we also want to preserve the local structure among the entities and the keywords.

3.2.1 Minimizing the distance

Let X and $H \in \mathbf{R}^{\mathbf{n \times d}}$ be matrices representing embeddings x_i and h_i respectively. We want to learn

$$u \in \mathbf{R}^d = \underset{u}{\operatorname{argmin}} \|(X - H)u\|_2$$

However, this leads to trivial solution with $u = 0$. To deal with this, we put the constraint $u^T u = 1$. Hence, the final objective can be described as :

$$u = \underset{u}{\operatorname{argmin}} \|(X-H)u\|_2^2 \text{ subject to } u^T u = 1$$

3.2.2 Preserving the local structure

It's known that maximising the variance while learning the lower dimensional embeddings also preserves the local structure between the embeddings. This would be similar to the objective function obtained while doing Principal Component Analysis (PCA). More specifically, $\max(var(Xu)) = \max(u^T var(X)u)$. However, this leads to $u = \infty$. We bound u by imposing the condition $u^T u = 1$. A similar approach follows for matrix H. So, the objective function is written as:

$$u = \underset{u}{\operatorname{argmax}}(u^T \Sigma_x u + u^T \Sigma_h u)$$
$$\text{subject to } u^T u = 1.$$

where, Σ_x and Σ_h are covariance matrices of X aand H respectively.

3.2.3 Combined loss function and Primal

It's difficult to combine both these loss functions, since they are both convex in u, but in one case we want to maximise the objective whereas in another case, we want to minimise it. One way to do this could be the following:

$$u = \underset{u}{\operatorname{argmin}} \|(X-H)u\|_2^2 - u^T(\Sigma_x + \Sigma_h)u$$
$$\text{subject to } u^T u = 1$$

However, the above function won't be convex. An alternative formulation could be :

$$u = \underset{u}{\operatorname{argmin}} \frac{\|(X-H)u\|_2^2}{u^T(\Sigma_x + \Sigma_h)u}$$
$$\text{subject to } u^T u = 1$$

This objective now looks very similar to what we observe in the Fisher's Discriminant Analysis(FDA). Inspired from FDA, we can modify the above mentioned objective and constraints to :

$$u = \underset{u}{\operatorname{argmin}} \|(X-H)u\|_2^2$$
$$\text{subject to } u^T(\Sigma_x + \Sigma_h)u = 1$$

This is the primal problem for our hypothesis. We can find the solution to this using primal-dual formulation or QP based methods.

3.3 Dual Formulation

The objective of the function can be re-written as

$$\|(X-H)u\|_2^2 = u^T(X-H)^T(X-H)u$$

We will start by writing the Lagrangian of the problem. Let's assume for $\nu \in R$

$$\mathcal{L}(u,\nu) = u^T(X-H)^T(X-H)u + \nu[u^T(\Sigma_x + \Sigma_h)u - 1]$$

Simplifying further, $\mathcal{L}(u,\nu)$ is given by

$$= u^T[(X-H)^T(X-H) + \nu(\Sigma_x + \Sigma_h)]u - \nu$$

The Lagrange dual function is defined as:

$$g(\nu) = \inf_u \ \mathcal{L}(u,\nu)$$

Lagrangian is a quadratic function, and is unbounded below if the minimum eigen value of the matrix $(X-H)^T(X-H) + \nu(\Sigma_x + \Sigma_h)$ is negative.

Using these properties of eigen values, the dual function can be given by

$$g(\nu) = \begin{cases} -\infty, & \text{for } \lambda_{\min}[(X-H)^T(X-H) \\ & \quad +\nu(\Sigma_x + \Sigma_h)] < 0 \\ -\nu, & \text{otherwise} \end{cases}$$

Here, the $\lambda_{\min}$ denotes the smallest eigen value of the matrix given. The dual problem is hence given by

$$\max_\nu \ g(\nu)$$

where, ν is constrained in the piece-wise function defined above.

3.4 KKT conditions

We have differentiable objective and constraint functions. Let u^* and ν^* be the optimal primal and dual solutions. So, the KKT conditions (Boyd and Vandenberghe, 2004) for strong duality that holds are mentioned below:

- $\nabla_u \mathcal{L}(u^*, \nu) = 0$

  $$2[(X-H)^T(X-H) + \nu(\Sigma_x + \Sigma_h)]u^* = 0$$
  $$\implies (X-H)^T(X-H)u^* = -\nu(\Sigma_x + \Sigma_h)u^*$$
  $$\implies (\Sigma_x + \Sigma_h)^{-1}(X-H)^T(X-H)u^* = -\nu\, u^*$$

  This behaves like a generalized eigenvalue problem (Ghojogh et al., 2019). Since the objective is minimization, our optimal primal solution will be the eigenvector corresponding to the smallest eigenvalue of the matrix $(\Sigma_x + \Sigma_h)^{-1}(X-H)^T(X-H)$. It is to be noted that $(\Sigma_x + \Sigma_h)$ is the sum of two covariance matrices which should be invertible.

- $(u^*)^T(\Sigma_x + \Sigma_h)u^* - 1 = 0$

- We don't have any inequality constraints, hence, the complimentary slackness conditions are not needed. Similarly, the feasibility conditions related to inequality conditions are not needed.

4 Dataset

We are using a supervised NER dataset (Ding et al., 2021) which contains 188,239 labeled sentences with its corresponding entities. It is further divided into 131,767 training samples, 18,824 samples in dev-set, and 37,648 samples in test set. Each word can be labeled into one of 8 possible coarse entities. Further, we aim to learn the mapping of a word into 66 possible fine grained classes. We want to emphasize the fact that we don't use fine grained labels while training.

5 Approaches

The entire problem is divided into stages where each stage performs a specific task in our method. This section elaborates on each step in depth.

5.1 NER Training and Keyword Generation

- **Training a classifier for NER**: We use a transformer based pretrained model - RoBERTa (Liu et al., 2019) to classify the entities into 8 coarse grained labels.

  The sentence is preprocessed and tokenized before passing as input to the model, which predicts the probability for each token in the sentence. The most probable class for each word is taken as the predicted entity. The model is trained on 131,767 sentences and evaluated on the test set, which is filtered before training the model.

- **Explaining predictions**: For every entity that the classifier predicts into a particular class, we use LIME (Ribeiro et al., 2016) to get the importance of keywords that triggered the classification. We get a rank of each keyword relevant to its classification for every entity. We keep the top-ranked keyword for generating the H matrix defined in the primal. Another transformed-based model is used to rerank the keywords for a certain entity type to filter out keywords that are not pertinent. A list of keywords corresponding to each entity can be found in Table 1

5.2 Generating the matrices

Once we have the keywords tagged to each entity, the following steps help generate the matrices X and H used in the primal.

- **Embedding the words:** We use a pretrained model to get the embeddings of entities and keywords in each sentence. The model takes the sentence as input and generates 768-dimensional embeddings for each token. Entity embeddings are pushed to X and corresponding keyword embeddings are pushed to matrix H.

- **Representing fine-grained classes:** Each fine NER label is appended with its corresponding coarse label to form a sentence and generate average embeddings for fine classes. A 768 dimensional vector will again represent the embedding of a class.

- **Projecting the matrices into low dimensional manifold:** The matrices X and H have 768 dimensions which seem to generate covariance matrices with very high variance. We utilize PCA (F.R.S., 1901) to project X and H matrices into a 20 dimensional manifold. The representation for fine-grained classes is also transformed in conjunction with matrix H.

5.3 Solving the Optimization Problem

After the desired representation matrices X and $H$are found, we solve the problem according to the KKT conditions.

Since the solution to the problem involves inverting the sum of the two covariance matrices, we also experiment by adding a slight noise to diagonal entries of $\Sigma_x + \Sigma_h$.

The number of eigenvectors to project the generated matrices is a hyperparameter, and we perform experiments with the different number of components.

5.4 Final prediction

For the final prediction for a given entity, we find the embedding of the class nearest to the entity embedding in the newly learnt projected space.

5.5 Experimental details

The constraints in our primal contains the sum of two covariance matrices, however, we also experiment with multiple variants of this formulation where the inherent solution stays the same, but the matrices are changed. These variations include utilizing the class embeddings and keyword covariance matrices in isolation. We also perform some experiments by utilizing kernel methods on

Coarse Entity	Ranked Keywords
B-location	trail, parkway, lake, village, island, city, coast, valley, country, coast, state, province
B-organization	team, paramilitary, football, media, subsidiary, football, conservative, professional
B-product	ship, game, aircraft, fitted, prototype, compatible, system
B-person	politician, senator, member, chief, director, sergeant, artist, poet, friend, president
B-other	language, disease, gene, award, species, use, portion, existence, syndrome
B-event	tournament, annual, knockout, September, last, second, inaugural, ongoing
B-art	film, music, drama, written, book, hit, song, cast, debut, comedy, album
B-building	clinic, house, airport, hospital, facility, campus, opening, restaurant, laboratory

Table 1: The table shows the list of keywords found by our model for each entity. It is evident that the model is able to generate relevant words

Precision	Recall	F1
0.28	0.31	0.29

Table 2: Precision, Recall and F1 score obtained for fine grained entity classification with weak supervision

K	10	50	100	200	300
P@K	0.4	0.36	0.38	0.34	0.34

Table 3: P@K obtained for fine grained entity classification

matrices X and H and the future possible work is discussed in Section 7.

Training Time: We performed training on Nvidia GPU. It took 3 hours to perform initial training and finding the keywords. The remaining section of the pipeline requires finding eigenvectors of d*d matrix which can be done within 5 minutes since the value of d is < 1000 and matrices are symmetric. Finding eigenvalues and nearest neighbors are executed in less then 5 minutes.

6 Results

In this section, we discuss the quantitative results and the qualitative results from our experiments.

6.1 Quantitative results

Following the convention of reporting the results for the NER task, we choose to report the precision, recall, and F1 score. Since there are 66 classes and each class might have a different number of samples, we report weighted scores for all the metrics in Table 2. We observe a weighted F1 score of 0.29.

Although this score is less than what is typically achieved in supervised settings with deep learning approaches ($\sim$ 0.7) (Ding et al., 2021), it can be noted that we train our models in low resource scenario with weak supervision- hence the comparison is not apt. In similar settings with weak or distant supervision, albeit for different tasks such as relation extraction, the reported F1 scores have been close to 0.4. (Hoffmann et al., 2011; Zeng et al., 2015) We should be able to improve upon the current F1 score even in this framework with different variations of the formulation, and careful feature engineering. We have seen incremental gains through these modifications.

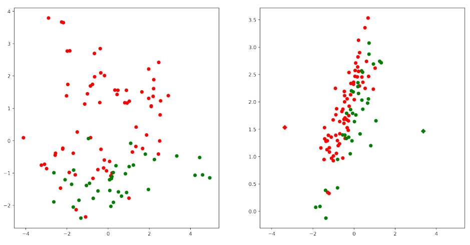

Figure 1: Keyword embeddings for top keywords of the sub-classes *actor* and *scholar*. The image on left is embedding of keywords initially and image on right shows embeddings of keywords after the transformation.

Another used metric for such data mining tasks is $P@K$. Out of the most confident K predictions made by the model, find the number of correct predictions. We report the value obtained for the metric $P@K$ for different values of K in the table 3. We observe that the value of $P@K$ decreases with increasing values of K. This is expected since the samples predicted with higher confidence would be more likely to be predicted correctly. This score is calculated on the test subset represented by the extracted entities.

6.2 Qualitative results

We discuss a few top confident samples predicted by the model. In the sentence, *"Due to the premature end of the series, the plans of actor Himmanshoo playing dual role and the entry of Kumar were dropped"*, the keyword learned is *actor*. Both the entities (denoted by "Kumar" and "Himanshoo") are correctly classified as *person-actor* by the final model. For another sentence, *"Klaus (29 October 1925 – 10 November 2015) was a German-born mathematician who won the Fields for proving Roth 's theorem on the Diophantine approximation of algebraic numbers"*, the model correctly learns that the first entity "Klaus" is a *person-scholar*. The keyword learnt in the sentence is *"mathematician"* which is also a representative of the class *"scholar"*.

We also discuss one failure case to bring some light to the shortcomings of the model. In the sentence *"The film also featured the voice of Walter as the reminiscing grown Sterling , Steve as his father Willard and Pamela as his sister Theo"*, the keyword selected for the entity *"Walter"* is sister which is possibly a correct keyword for the class *"person"* but not representative of the sub-class *"person-actor"*.

In the figure 1, we show the keyword embeddings for top keywords of the sub-classes *actor* and *scholar*. We observe that for the current two classes, the keywords embeddings cluster better. However, in most cases where keywords are common between subclasses, the clustering is not clear.

7 Conclusion and Future Work

In our work, we trained the NER on coarse labeled dataset and extracted keywords responsible for the classification. Then, we proposed a framework to learn the representation of keywords and entities, and perform fine grained classification based on those representations. We report F1 score, and Precision@K metrics for our test subset represented by the extracted entities. We also perform a qualitative study to understand the success and shortcoming of our method.

In future, we plan to experiment extensively with learning the eigenvectors in a high-dimensional Hilbert Space ($\mathcal{H}$). The idea is to use a radial basis function kernel which projects the matrix X and H into a infinite dimensional space. The optimization problem should hold even in those spaces, and we can use kernel functions to project on the resultant eigenvectors. Due to time constraints, we couldn't do extensive experimentation and would also like to do more further experiments with the current formulation.

References

Parul Awasthy, Taesun Moon, Jian Ni, and Radu Florian. 2020. Cascaded models for better fine-grained named entity recognition.

Stephen Boyd and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. Few-NERD: A few-shot named entity recognition dataset. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3198–3213, Online. Association for Computational Linguistics.

Karl Pearson F.R.S. 1901. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

Benyamin Ghojogh, Fakhri Karray, and Mark Crowley. 2019. Eigenvalue and generalized eigenvalue problems: Tutorial. *arXiv preprint arXiv:1903.11240*.

Sonal Gupta and Christopher Manning. 2014. Improved pattern learning for bootstrapped entity extraction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 98–108, Ann Arbor, Michigan. Association for Computational Linguistics.

Nguyen Van Hoang, Soeren Hougaard Mulvad, Yuan Rong Dexter Neo, and Yang Yue. 2021. Zero-shot learning in named-entity recognition with external knowledge. *CoRR*, abs/2111.07734.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 541–550, Portland, Oregon, USA. Association for Computational Linguistics.

Jiacheng Li, Haibo Ding, Jingbo Shang, Julian McAuley, and Zhe Feng. 2021a. Weakly supervised named entity tagging with learnable logical rules.

Jiacheng Li, Haibo Ding, Jingbo Shang, Julian J. McAuley, and Zhe Feng. 2021b. Weakly supervised named entity tagging with learnable logical rules. *CoRR*, abs/2107.02282.

Pierre Lison, Aliaksandr Hubin, Jeremy Barnes, and Samia Touileb. 2020. Named entity recognition without labelled data: A weak supervision approach.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30:3–26.

Cheng Niu, Wei Li, Jihong Ding, and Rohini Srihari. 2003. A bootstrapping approach to named entity classification using successive learners. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 335–342, Sapporo, Japan. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier.

Esteban Safranchik, Shiying Luo, and Stephen H. Bach. 2020. Weakly supervised sequence tagging from noisy rules. In *AAAI*.

Xuan Wang, Xiangchen Song, Bangzheng Li, Kang Zhou, Qi Li, and Jiawei Han. 2020. Fine-grained named entity recognition with distant supervision in covid-19 literature. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 491–494.

Vikas Yadav and Steven Bethard. 2019. A survey on recent advances in named entity recognition from deep learning models. *CoRR*, abs/1910.11470.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762, Lisbon, Portugal. Association for Computational Linguistics.