APCNN: Tackling Class Imbalance in Relation Extraction through Aggregated Piecewise Convolutional Neural Networks

Alisa Smirnova
Exascale Infolab
University of Fribourg
Fribourg, Switzerland

Julien Audiffren
CMLA, ENS Paris Saclay
University Paris Saclay
Cachan, France

Philippe Cudré-Mauroux
Exascale Infolab
University of Fribourg
Fribourg, Switzerland

***Abstract*—One of the major difficulties in applying distant supervision to relation extraction is class imbalance, as the distribution of relations appearing in text is heavily skewed. This is particularly damaging for the multi-instance variant of relation extraction. In this work, we introduce a new model called Aggregated Piecewise Convolutional Neural Networks, or APCNN, to address this problem. APCNN relies on the combination of two neural networks, a novel objective function as well as oversampling techniques to tackle class imbalance. We empirically compare APCNN to state-of-the-art approaches and show that it outperforms previous multi-instance approaches on two standard datasets.**

I. Introduction

Relation extraction is an essential part of the effort towards making natural language text machine-readable and processable. One of the main current limitations in applying machine learning to relation extraction is the lack of training data. To address this problem the *distant supervision* paradigm was proposed by [1]. Distant supervision allows to automatically label large text corpora using knowledge bases as a source of supervision instead of relying on humans to manually annotate data (which is both complex and costly in practice). More specifically, given a text corpus $\mathcal{T}$ and a knowledge base $\mathcal{D}$, distant supervision identifies all sentences that mention a given entity pair (e_1, e_2) and label them with a relation r if there exists a corresponding triple (e_1, r, e_2) in the knowledge base, and with the *None* relation otherwise (specifying the fact that two entities are not linked by any relation). Sentences sharing the same entity pair are called a *bag*. Such bags are then used to train a model which aim to predict relations for previously unseen pairs of entities appearing in a sentence.

Unfortunately, the distribution of labels obtained through distant supervision is typically highly skewed. The vast majority of labels corresponds to the None relation, for two main reasons: i) often, two entities are co-located in text because they are sharing some context without explicitly taking part of a relation; such sentences are likely to be labeled with None and ii) all knowledge bases are incomplete (i. e., many facts are missing), resulting in many pairs of entities erroneously labeled as None despite taking part of a more specific relation in text. Moreover, some relations are more frequent than others in text (see for instance Table I). Therefore, there is a significant class imbalance in the data. This often leads to classifiers achieving high performance on some metrics, such as accuracy, but which exhibit a low discriminative power in practice (a common problem in class imbalance). That is, the classifier that assigns the label of the most frequent class has high accuracy, despite being impractical.

TABLE I
Example of relations labels and their frequency in the training and test datasets from the New York Times corpus [2].

Relation	% in training set	% in test set
None	91.21%	89.32%
/business/company/founders	0.09%	0.13%
/business/person/company	0.71%	1.00%
/location/location/contains	3.73%	4.68%
/people/person/place_of_birth	0.67%	0.03%

In this work, we consider two types of class imbalance: None vs not-None and inequality between specific relations. For instance, in the New York Times corpus [2] the relation /location/location/contains is 43 times more frequent than /business/company/founders (see Table I). We tackle both types of class imbalance by dividing the component responsible for relation extraction into two: a first subcomponent performing binary classification separating None and not-None classes and a second one predicting a particular relation label. For both components, we use a *resampling* technique that allows to compensate for underrepresented classes.

We solve both tasks jointly by introducing APCNN – an aggregated piecewise convolutional neural network that tackles both the task of identifying the entity pairs that are not related (i. e., *None* label) and estimates the likelihood of relation labels. Furthermore, we introduce a new bag-level loss function based on the Ordered Weighted Averaging operators (OWA; see e. g., [3]).

We compare our approach to two state-of-the-art approaches: piecewise convolutional neural networks (PCNN) [4] and CoType, a joint entity and relation extraction model [5]. We show that our approach outperforms PCNN both in distinguishing between None and not-None classes and in predicting specific relation labels. A major

shortcoming of CoType is a low precision on predicting the presence of a relation – it often classifies None relations as not-None – a problem that is not encountered by APCNN.

II. Related Work

a) Distant Supervision: Distant supervision was originally introduced by [1] to overcome the lack of training data in automatic relation extraction. Many improvements of the original approach were proposed to mitigate the impact of the two main sources of noise that occurs in automatically labeled data: i) sentences mentioning two related entities while not expressing a relation between them and ii) incompleteness of the knowledge base, resulting in some sentences being wrongly labeled with None (see [6] for a comprehensive survey on this topic).

b) Word Representations: In contrast to the most of the approaches on relation extraction (see [6]), our model does not use lexical or syntactic features. Instead, we rely on the skip-gram model [7] – a distributional word representation that was shown to achieve competitive performance (see [4], [8], particularly [9] for relation extraction).

c) Neural Relation Extraction: [10] introduced a convolutional neural network (CNN) model for sentiment analysis, where the authors showed that models using pre-trained non-static embeddings lead to the best performance. Following this work, [4] proposed a CNN-based model for relation extraction. In addition to the word embeddings, their model used *position features* to encode the relative distance between a given word and the entity of interest. Since with distant supervision only the label of the bag is known and the label of the particular sentence remains unknown, the authors introduced a bag-level loss function to overcome this problem by taking into account only the sentence that maximizes the likelihood of the correct label for each bag. This model has the drawback of only using information for the *most likely* sentence of the bag. [11] and [12] proposed a solution to this problem by introducing a selective attention mechanism over sentences in the bag. Our model, APCNN, uses a similar CNN architecture as one of its components. However, APCNN greatly differs from previous work as i) it relies on a non-linear combination of two different networks, ii) it uses a new bag-level loss function that is better suited for class imbalance, and also accounts for each sentence in the bag (see Section III-C) – an alternate approach to the attention mechanism used in [11] – and iii) it uses resampling techniques. As shown in Section IV, our approach successfully addresses class imbalance and can discriminate between a set of relations with a high accuracy, while previous neural network based models are heavily biased towards predicting *None* most of the time.

d) Class Imbalance: As noted in [8], relation extraction is often applied on strongly imbalanced datasets. Class imbalance between the different relations significantly impacts the performance of most learning algorithms [13]. A number of approaches have been developed to address this issue, including methods that modify the training set or change the loss function (see [14] for an in-depth review). In this paper, we combine several of these tools with APCNN to address the class imbalance problem, including a resampling tool called oversampling (a method that modify the training set to balance classes [15]), a weighted loss function (see e. g., [16]) and a divide and conquer approach that splits the original problem into more balanced subproblems, whose solutions are then combined with classifier aggregation [17]. To the authors' knowledge, the combination of methods introduced in this work (particularly network aggregation) is a novel way to address the class imbalance problem in relation extraction.

III. Methodology

In this section, we describe our approach to address the problem of class imbalance in relation extraction. We introduce a new learning algorithm, APCNN, based on Piecewise Convolutional Neural Network (PCNN), originally described in [4]. We additionally propose multiple modifications to the learning process in order to accommodate the strongly unbalanced classification problem, including:

- a new neural network architecture aggregating two PCNNs, the first one predicting the absence of relation, and the second one estimating the likelihood of each relation;
- the processing of the dataset through resampling techniques. Different resampling techniques are applied to each of the two aforementioned PCNNs;
- a new loss function, based on the Ordered Weighted Averaging operators (OWA; see e.g. [3]), that improves over previous bag-level loss functions.

A. Aggregated PCNN

Our network, APCNN (Aggregated PCNN), is a non-linear combination of two PCNNS. It relies on three hyperparameters (τ, ε and λ – we describe them below). We start by briefly presenting the main characteristics of PCNNs below; we refer a reader to [4] for more detail.

a) Structure of a PCNN: PCNNs consist of five main components: bag of sentences, embedding layer, convolutional layer, piecewise max-pooling and softmax classifier.

b) Bag of Sentences: In our context, training set consists of bags of sentences, or bags for short. Bags are labeled with a pair of entities and a relation, and are made of multiple sentences mentioning those two entities. While the relation is known at bag-level, the exact relation expressed in each sentence individually is unknown. The bag approach relies on the *at-least-one* assumption [18], i. e., at least one sentence in the bag expresses the aforementioned relation.

c) Embedding Layer: In a PCNN, each input word of a given sentence is encoded using an embedding matrix, which is updated during training. The matrix can be either randomly initialized or pre-trained on an external dataset. Positional features – which capture the relative position of the current word with respect to the two entities of interest in the sentence – are then added to the word embedding, resulting into the word encoding.

d) Convolutional Layer and Piecewise Max-pooling: A one-dimensional convolutional layer is applied to the embedding of each sentence in the bag, followed by a piecewise max-pooling layer (see [4] for an in depth discussion on the strengths of CNN in relation extraction). Compared to a regular max-pooling, piecewise version extracts three different values of each filter activation signal: one for n-grams containing at least one word before the first entity (the first "piece" of the sentence), one for n-grams containing words between the two entities, and a third one for the remaining n-grams. As a result, three features are extracted for each filter on each sentence.

e) Softmax Layer: Finally, the resulting features for each filter are combined with a softmax layer in order to predict the probability of each relation.

f) Contribution: APCNN: In this work, we consider "None" as a "special" relation, as i) it is the most frequent relation by a large margin and ii) it only reflects the absence of a relation.

Following this intuition, we decided to handle "None" relations separately from the others by training two different PCNNs:

- A Binary classifier (BClass) that predicts whether an entity pair has some relation, therefore learning to identify the "None" relation;
- A Multiclass classifier (MClass) that for a given bag predicts the exact relation label.

The resulting network architecture is summarized in Figure 1 and works as follows. Given a sentence in a bag, it computes the encoding of the sentence using a common embedding layer. The resulting signal is then fed into two independent Convolutional - Piecewise Max-Pooling - Softmax sequences of layers that produce respectively the probability of the "None" relation $p_{\text{None}} \in [0, 1]$ (BClass) and the probability of each other relation $\left(p_i\right)_{i=1}^{n}$ (MClass). APCNN then predicts $\mathbf{p}$, the probability of each class, as follows:

$$\mathbf{p}(\text{None}) = \begin{cases} p_{\text{None}} & \text{if } p_{\text{None}} > \tau, \\ \varepsilon & \text{otherwise.} \end{cases} \tag{1}$$

$$\mathbf{p}(i) = \begin{cases} p_i(1 - p_{\text{None}}) & \text{if } p_{\text{None}} > \tau, \\ p_i(1 - \varepsilon) & \text{otherwise.} \end{cases} \tag{2}$$

where τ and ε are hyperparameters. It is easy to see that $\mathbf{p}$ defines a probability distribution over all possible relations (including None).

Our experiments illustrate that the non-linear combination of BClass and MClass provides more flexibility to APCNN – as it is able to both predict None and distinguish between relations successfully (see Section IV for more details).

g) Encoding Sharing: While BClass and MClass have different objectives – and therefore different filters and fully connected layers – they both attempt to predict relations (or the lack thereof) between entities. As such, parts of the learned features and embeddings from each network may contain information useful for the other one. Following this idea, we constrain BClass and MClass to use the same encoding matrices. This constraint forces the sharing of embeddings – and therefore learned information – between the two PCNNs.

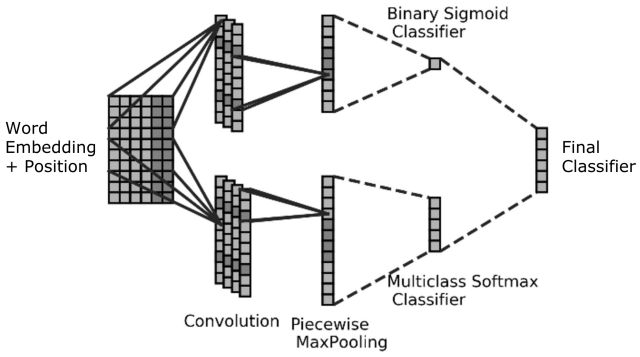

Fig. 1. The architecture of APCNN, illustrating the behavior of the network on a sentence. Both convolution layers share the same inputs. The final classifier is the combination of the two probabilities produced by the PCNNs.

B. Resampling

While APCNN attempts to address the problems raised by the None relation, there is typically significant class imbalance among regular relations as well (see Table I for instance). To alleviate this problem, we use a resampling technique during the training steps of the network.

Resampling techniques – and oversampling in particular – have been successfully used to address the class imbalance problem in neural networks [19]. Therefore, we combined APCNN with the oversampling of the infrequent classes. Since oversampling should be used carefully, particularly in datasets with many different classes, we only partially oversample the least frequent classes (see Section IV-C), and address the remaining class imbalance using weighted loss (see below).

C. Relaxed Bag-level Loss Function

In [4], the authors used a bag-level loss function that relies on $\arg\max$: only the best sentence in the bag – i. e., the sentence that predicts the true relation with the largest probability – is part of the cost. While this approach has significant advantages compared to regular loss – such as a better embedding of the *at-least-one* hypothesis – the use of $\arg\max$ has several downsides, including a gradient that only updates one sentence at a time. To train APCNN, we use a new bag-level loss function. Let $0 \leq \lambda < 1$ be the *relaxation* coefficient (a hyperparameter of the model). Let $\mathcal{B} = \{s_1, \dots s_n\}$ be a bag of sentences labeled with the relation r, and $\mathbf{p}(r|s_i)$ be the probability of the relation r predicted by APCNN from sentence s_i. We define p_{loss} – the ordered weighted average (OWA) of the probabilities of the sentences of $\mathcal{B}$ – as follows:

$$p_{\text{loss}}(r|\mathcal{B}) = (1 - \lambda) \max_{s \in \mathcal{B}} \mathbf{p}(r|s) + \frac{\lambda}{n} \sum_{s \in \mathcal{B}} \mathbf{p}(r|s), \tag{3}$$

and the corresponding loss function $\mathcal{J}$ as:

$$\mathcal{J}\left(\mathcal{B}\right) = -w_r \log\left(p_{\text{loss}}(r|\mathcal{B})\right), \tag{4}$$

where w_r is a weight that is inversely proportional to the proportion of class r *after* resampling. Note that with $\lambda = 0$

TABLE II
COMPARISON OF BINARY CLASSIFICATION. PRECISION AND RECALL ON HIGHEST F1-SCORE.

Model	Dataset	Precision	Recall	F1	Weighted Accuracy	AUC
APCNN	NYT	0.44	0.42	0.43	25.74 %	0.78
PCNN	NYT	0.38	0.42	0.4	13.47 %	0.76
CoType	NYT	0.1	1.0	0.19	46.03 %	0.38
APCNN	Wiki	0.36	0.38	0.37	77.70 %	0.82
PCNN	Wiki	0.28	0.58	0.38	60.58 %	0.83
CoType	Wiki	0.24	0.46	0.32	85.43 %	0.73

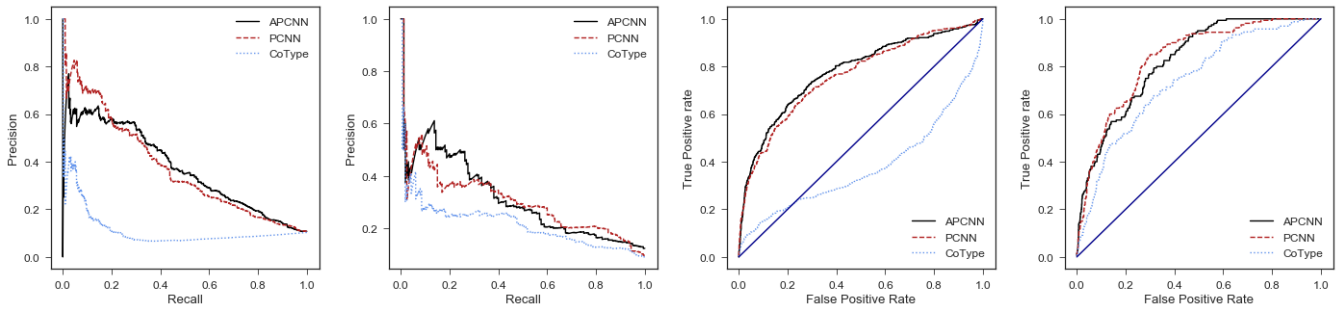

Fig. 2. From left to right: Precision/Recall curves (NYT, Wiki-KBP) and ROC curves (NYT, Wiki-KBP) for the existence of a relation – i. e., the prediction of the None relation – for APCNN (black), PCNN (red), and CoType (blue). APCNN performs slightly better than PCNN on this task, and significantly better than CoType.

and $w_r = 1$, (4) can be reduced to the bag-level loss introduced in [4]. The w_r are used to alleviate the class imbalance that remains after the resampling steps. λ is used to transform the $\arg\max$ used in [4] into a more general OWA. It has been shown that slight relaxations of the $\arg\max$ (i. e., small positive values of λ) lead to better theoretical properties and numerical stability (see [3] and references therein). Additionally, the use of $\lambda > 0$ in (3) extends the gradient updates to all the sentences of the bags, slightly increasing the convergence speed.

IV. EXPERIMENTAL EVALUATION

The purpose of the following experiments is to highlight the class imbalance problem and its impact on relation classifiers, while illustrating the benefits and drawbacks of APCNN. We extensively compare our approach to PCNN [4] – the network from which APCNN was initially derived – and CoType [5] – a state-of-the-art approach that relies on additional semantic features – on several datasets with respect to multiple metrics.

A. Hyperparameter Tuning

CoType and PCNN hyperparameters were chosen following the recommendations of [4] and [5]. For APCNN, we used a 50 dimensional word embeddings, combined with two 5 dimensional positions features, resulting in a 60 dimensional encoding. For both BClass and MClass, we used 230 filters of size 3 for their convolutional layers, i. e., the parameters recommended by [4]. Additionally, we selected ε, λ and τ by cross validation – resulting in the choice $\varepsilon = 0.01$, $\lambda = 0.01$ and $\tau = 0.3$ which gave the best overall performance. It is worth noting that APCNN metrics are robust to small changes of these values.

B. Metrics

In order to properly study class imbalance, we first need to introduce additional metrics. Indeed, usual evaluations such as precision, recall or accuracy are ill-suited for significantly imbalanced datasets. For instance, a classifier that always predicts the None relation achieves very high accuracy (around 90%, see Table I). Therefore, we decided to use three additional metrics: AUC, confusion matrix and weighted accuracy.

- **Receiving Operator Curve (ROC) and Area Under ROC (AUC).** This commonly used statistical tool measures the evolution of true positives with respect to false positives. While aimed at binary classification, this operator possesses a number of interesting properties, and is in particular robust to class imbalance [20].
- **Confusion Matrix.** A confusion matrix encodes for each pair of relations r_i, r_j the percentage $m_{i,j}$ of elements of r_i predicted as r_j. Hence, the confusion matrix offers a comprehensive summary of the performance of the classifier on each class.
- **Weighted Accuracy** (wacc). We also consider the weighted accuracy defined as follows:

$$\text{wacc} = \frac{1}{|R|} \sum_{r \in R} \frac{\#\text{correct predictions of } r}{\#\text{samples labeled with } r},$$

where R is the set of relations. To achieve a large wacc, a classifier needs to correctly predict the elements of each class, and not only the most frequent one – in fact,

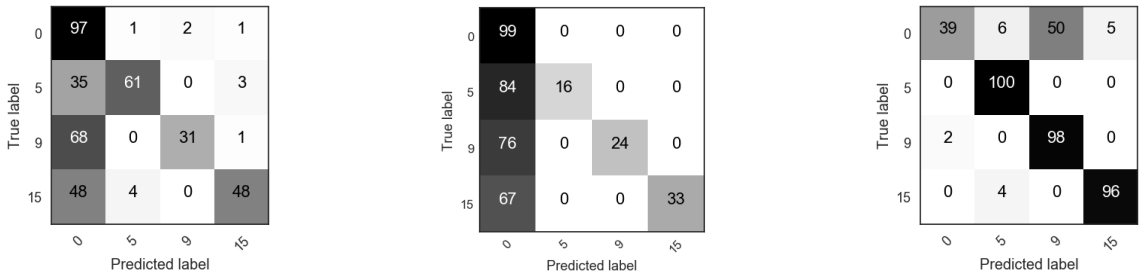

Fig. 3. Normalized confusion matrices of APCNN (left), PCNN (center) and CoType (right) for the relations 'None','/business/person/company','/location/location/contains' and '/people/person/nationality' in the NYT dataset. The element i, j of each matrix indicates the percentage of elements of class i predicted as j.

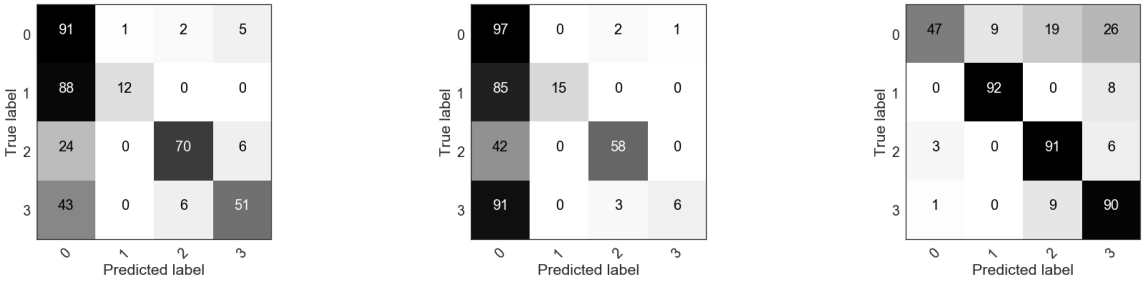

Fig. 4. Normalized confusion matrices of APCNN (left), PCNN (center) and CoType (right) for the relations 'None', 'per:children', 'per:country_of_birth' and 'per:country_of_death' in the Wiki-KBP dataset. The element i, j of each matrix indicates the percentage of elements of class i predicted as j.

an algorithm that only predicts the most frequent class achieves a weighted accuracy of $1/R$.

C. Dataset and Data Preprocessing

We evaluate our approach on i) the New York Times corpus [2] – which was originally prepared for relation extraction by aligning it to Freebase[1] in [18] and ii) the Wiki-KBP dataset [21][2].

As discussed in Section III-B, for each dataset we oversample the infrequent classes by adding copies of randomly selected elements until the ratio between the least frequent class and the most frequent class is larger than 1/5.

D. Noisy Labels

One of the shortcomings of distant supervision is noisy labels. That is, the entity pair can be erroneously labeled with None because of knowledge base incompleteness, or a sentence may not express the relation that it is labeled with. For instance, the following sentence:

> "It is **Wartburg College** in Waverly, **Iowa**, not the University of Wisconsin-River Falls "

is labeled with the relation None for the entity **Wartburg College** and **Iowa** in NYTimes test set. This false labeling could induce a wrong evaluation of the true performance of the different algorithms. Therefore, we carefully checked the label of the test set for both datasets using multiple knowledge bases (Freebase, DBpedia and Wikipedia infoboxes) while not modifying the training set, even for wrongfully labeled sentences.

[1]https://developers.google.com/freebase/

[2]For both datasets we used their versions preprocessed by [5], including the train/test split

E. Results

We use two different tasks to properly compare the APCNN, PCNN and CoType.

a) Predicting None: First, we evaluate the performance of the three models in recognizing the presence of a relation – in other words, at identifying the None relation. This is important due to the particular nature of None and its prominence in the class distribution, in both datasets. In order to do so, we compare for each sentence of the test set the output of the algorithm and the true label, and mark the prediction as true if both are either None or any relation but None. This reduces the problem to a binary classification problem; precision/recall curves for each algorithm are shown in Figure 2, and results are reported in Table II. In this experiment, APCNN performs similarly to PCNN, improving metrics by a slight amount for the NYTimes dataset, while being slightly behind for the Wiki-KBP dataset. Conversely, CoType significantly underperforms the other two algorithms on this task on both cases; it is important to mention however that this framework differs from [5], where the algorithm dealt with entity detection, relation detection and relation labeling.

b) Distinguishing between relations: Second, we analyze the performance of the algorithms over all relations, including None. The weighted accuracy is reported in Table II. As there are many different relations in each datasets, we only report

part of the confusion matrices in Figure 3 (NYTimes) and 4 (Wiki KBP). In this experiment, APCNN achieves a much greater weighted accuracy than PCNN (with a value *twice* higher for the NYTimes dataset), highlighting the advantage of our approach. Indeed, this difference in performance is largely due to the fact that APCNN is able to learn patterns regarding the different relations, despite the strong class imbalance, while PCNN predicts None most of the time. On this task, CoType outperforms both PCNN and APCNN, illustrating the advantages of the numerous additional features that this model considers.

c) Ablation Analysis: Third, we conducted an ablation analysis of the different components APCNN by removing each of them (Improved loss function, Resampling, Dual Networks) one at a time. Unsurprisingly, all the elements appeared to contribute significantly to the performance of APCNN, and their removal drastically reduced its weighted accuracy.

F. Discussion

a) Confusion between similar relations: From the confusion matrix, it is interesting to note that APCNN frequently confuses relations that share similar characteristics. For instance, in the NYTimes dataset, "/location/location/contains" and "/location/country/capital" are geographic relations, and are more frequently confused by the model than unrelated labels such as "/location/location/contains" and "/business/person/company". This highlights the fact that APCNN is able to learn parts of the semantics of the different relations.

b) CoType and APCNN: While APCNN outperforms PCNN by better distinguishing between relations and by better predicting None, its comparison to CoType is more difficult. CoType achieves higher performance when distinguishing between relations, but is significantly underperforming for the relation detection task. Additionally, CoType uses multiple additional features, including lexical and grammatical information, while APCNN only uses an *unsupervised encoding* of the sentences. Overall, APCNN achieves a promising balance between the two tasks while only requiring little external information.

V. Conclusion

In this work, we introduced a new model called APCNN to address the class imbalance problem in relation extraction. APCNN significantly outperforms PCNN, the algorithm it was originally derived from, and strikes a good balance between predicting the existence of a relation on one hand and distinguishing between a set of known relations on the other hand. Future work might include the combination of APCNN and CoType, as they appear to exhibit complementary properties.

Acknowledgments

This project was supported by the Swiss National Science Foundation (grant #407540_167320 *Tighten-it-All*) and by the European Research Council (grant #683253 *GraphInt*).

References

[1] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 2009, pp. 1003–1011. [Online]. Available: http://www.aclweb.org/anthology/P09-1113

[2] E. Sandhaus, "The new york times annotated corpus," *Linguistic Data Consortium, Philadelphia*, vol. 6, no. 12, p. e26752, 2008.

[3] A. Emrouznejad and M. Marra, "Ordered weighted averaging operators 1988–2014: A citation-based literature survey," *International Journal of Intelligent Systems*, vol. 29, no. 11, pp. 994–1014, 2014.

[4] D. Zeng, K. Liu, Y. Chen, and J. Zhao, "Distant supervision for relation extraction via piecewise convolutional neural networks," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1753–1762. [Online]. Available: http://aclweb.org/anthology/D/D15/D15-1203.pdf

[5] X. Ren, Z. Wu, W. He, M. Qu, C. R. Voss, H. Ji, T. F. Abdelzaher, and J. Han, "Cotype: Joint extraction of typed entities and relations with knowledge bases," in *Proceedings of the 26th International Conference on World Wide Web,*, 2017, pp. 1015–1024. [Online]. Available: http://doi.acm.org/10.1145/3038912.3052708

[6] A. Smirnova and P. Cudré-Mauroux, "Relation extraction using distant supervision: A survey," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 106:1–106:35, 2019.

[7] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[8] T. H. Nguyen and R. Grishman, "Relation extraction: Perspective from convolutional neural networks," in *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, 2015, pp. 39–48.

[9] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *27th Annual Conference on Neural Information Processing Systems 2013*, 2013, pp. 3111–3119.

[10] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1746–1751. [Online]. Available: http://aclweb.org/anthology/D/D14/D14-1181.pdf

[11] Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun, "Neural relation extraction with selective attention over instances," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, vol. 1, 2016, pp. 2124–2133.

[12] G. Ji, K. Liu, S. He, J. Zhao *et al.*, "Distant supervision for relation extraction with sentence-level attention and entity descriptions." in *AAAI*, 2017, pp. 3060–3066.

[13] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent data analysis*, vol. 6, no. 5, pp. 429–449, 2002.

[14] S. Koco, "Tackling the uneven views problem with cooperation based ensemble learning methods," Ph.D. dissertation, Aix-Marseille, 2013.

[15] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[16] Y. Sun, M. S. Kamel, A. K. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern Recognition*, vol. 40, no. 12, pp. 3358–3378, 2007.

[17] R. Barandela, R. M. Valdovinos, and J. S. Sánchez, "New applications of ensembles of classifiers," *Pattern Analysis & Applications*, vol. 6, no. 3, pp. 245–256, 2003.

[18] S. Riedel, L. Yao, and A. McCallum, "Modeling relations and their mentions without labeled text," in *Machine Learning and Knowledge Discovery in Databases, European Conference*, 2010, pp. 148–163. [Online]. Available: https://doi.org/10.1007/978-3-642-15939-8_10

[19] Z.-H. Zhou and X.-Y. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 63–77, 2006.

[20] N. Vayatis, M. Depecker, and S. J. Clémençon, "Auc optimization and the two-sample problem," in *Advances in Neural Information Processing Systems*, 2009, pp. 360–368.

[21] J. Ellis, J. Getman, and S. M. Strassel, "Overview of linguistic resources for the tac kbp 2014 evaluations: Planning, execution, and results," in *Proceedings of TAC KBP 2014 Workshop, National Institute of Standards and Technology*, 2014, pp. 17–18.