NumER: A Fine-grained Numeral Entity Recognition Dataset*

Thanakrit Julavanich[1][0000−0003−1764−7794] and Akiko Aizawa[2,1][0000−0001−6544−5076]

[1] The University of Tokyo, Bunkyo, Tokyo 113-8654, Japan
[2] National Institute of Informatics, Chiyoda, Tokyo 101-8430, Japan
thanakrit@g.ecc.u-tokyo.ac.jp, aizawa@nii.ac.jp

Abstract. Named entity recognition (NER) is essential and widely used in natural language processing tasks such as question answering, entity linking, and text summarization. However, most current NER models and datasets focus more on words than on numerals. Numerals in documents can also carry useful and in-depth features beyond simply being described as cardinal or ordinal; for example, numerals can indicate age, length, or capacity. To better understand documents, it is necessary to analyze not only textual words but also numeral information. This paper describes NumER, a fine-grained **Num**eral **E**ntity **R**ecognition dataset comprising 5,447 numerals of 8 entity types over 2,481 sentences. The documents consist of news, Wikipedia articles, questions, and instructions. To demonstrate the use of this dataset, we train a numeral BERT model to detect and categorize numerals in documents. Our baseline model achieves an F1-score of 95% and hence demonstrating that the model can capture the semantic meaning of the numeral tokens.

Keywords: Named entity recognition · Numeral classification · Numeral understanding · Natural language understanding.

1 Introduction

Named entity recognition (NER) is an NLP subtask that identifies and locates an entity in unstructured text, then classifies that entity into a predefined category. NER is an essential part of many NLP tasks and applications such as question answering, entity linking, and text summarization [17]. Most NER models and datasets are designed to focus on word entities—that is, the entity token consists of alphabetical characters—such as those denoting people, locations, and organizations. However, there is only a limited set of categories available for numerals, in which the token consists of numerical characters. For example, the CONLL-2003 corpus has no numeral entity type [13], and the OntoNotes 5 corpus has the types Percent, Money, Quantity, Ordinal, and Cardinal [15].

* This work was supported by JST, AIP Trilateral AI Research, Grant Number JPMJCR20G9 and by NEDO, SIP-2 Program "Big-data and AI-enabled Cyberspace Technologies," Japan.

Equally as important as word tokens, numeral tokens also contain relevant information. Moreover, we need to categorize numeral entities in more detail than the current NER datasets can provide. In a real-life scenario, for example, in a biographical document, the text might include numerals describing the age, birth year, weight, and height of the subject of the biography. As shown in Figure 1, an NLP application such as a question-answering task could include an inquiry using a monetary numeral entity, e.g. "50" in "who ordered more than 50 USD worth of meat today", or population numeral entity, e.g. "50,000" in "where is the nearest stadium with a capacity of more than 50,000 people." Understanding these numerals may help the model to better determine the correct part of the article or the right property in the knowledge base. For example, when the model recognizes the abovementioned monetary token "50", it can focus on the monetary property, e.g. the order's payment amount.

Fig. 1: Examples of numeral entities.

Previous studies have performed some research about numeral entities. For example, Min et al. [8] presented a numeral classification method using a rule-based approach. However, the focused classes were both semantic category, e.g., Money and Date, and syntactic categories, e.g., Number and Floatnumber. As a result, there may be a conflict of category taxonomy in this scheme. For example, the token "22.23" in the context "22.23 USD" can be categorized as both Money and Floatnumber. Another related work is the NTCIR-14 FinNum Task [2]. The authors published a dataset for fine-grained numeral entity recognition in social media data from StockTwits. This work focused on the financial domain with finance-related entity types such as buy price, sell price, and stop loss.

To better understand fine-grained numeral information, we created a numeral taxonomy by classifying the answer's property from the existing datasets for question answering over tabular data and text-to-SQL semantic parsing. Because each question focuses on a small domain, the unit and object token can be easily omitted. Therefore, it provides a more difficult task for the model to classify. We sort numerals into eight categories: Age, Population, Year, Date/Month, Length/Height, Money, Weight/Volume, and Generic.

This paper focuses on numerals in unstructured text, rather than primarily targeting word tokens like typical NER datasets. This work's main objective is

to create an entity recognition dataset focused on numeral tokens. Moreover, we aim to ensure the model's ability to comprehend and capture the semantics of numerals before applying it to the downstream tasks. This work provides three main contributions. First, we annotate the numerals in the chosen text corpora with this taxonomy and construct a dataset for experiments. Second, we present the dataset focused on the cases where there is no unit token. Third, we conduct comprehensive investigations to compare the performance of different classification and entity recognition models. Our annotated dataset is published at `https://github.com/Alab-NII/ValER`.

2 Related Work

There are many approaches used to build NER systems, such as creating handcrafted rules [10] or using a machine learning model [18]. Furthermore, today, many NER resources are available for English and other languages. However, in typical NER models and datasets, the focus is on named entities, which are usually word tokens such as "The White House", which is an Organization entity, or "Taylor Swift," a Person entity. Because of this, NER is a beneficial tool to understand the words in sentences. However, there is still a limited number of works focused on numerals in NER.

2.1 Rule-based Numeral Entity Recognition

There have been several attempts to recognize or classify numeral tokens. Microsoft Recognizers Text[3] is an off-the-shelf tool for recognizing the numerals, units, and date/time expressed in documents. This library detects the unit token and matches pre-defined regular expression patterns to identify the numeral's type. For example, if the model detects the "USD" token, it can recognize the nearby numeral as belonging to the currency category. According to the numerals' units, it can detect numerals of the following four types, including Age, Currency, Dimension, and Temperature. However, if a sentence does not contain a unit token, recognition is difficult.

2.2 Machine Learning-based Numeral Entity Recognition

For the machine learning approach, the capability of the model is dependent on the dataset used for training, especially the target entity types. At present, the available entity types of numeral tokens are still limited. For example, in the OntoNotes dataset [15], there are seven entity types for numerals: Date, Time, Percent, Money, Quantity, Ordinal, and Cardinal. Although this design provides some understanding of numeral tokens, there is still room to extend this structure, especially regarding the Quantity class. Entity class extension can help understand measurements such as length, duration, volume, or number of items.

There have also been several datasets for numeral classification in a specific domain. For example, the NTCIR-14 FinNum task [2] focuses on numeral classification in informal financial documents. The focused entity types are finance-based, e.g., Quote, Change, Buy price, and Sell price. Several works have been

[3] `https://github.com/Microsoft/Recognizers-Text`

submitted to this shared task based on state-of-the-art and well-known language models and architectures, e.g., CNN with ELMo word embeddings [1], RoBERTa-based models [6], and multi-layer perceptrons with LSTM [16]. However, as these studies were tailored to specific domains, they cannot be effectively applied to more general cases.

3 The NumER Dataset

The NumER dataset is created using documents from several datasets. Each numeral is annotated and categorized into one of the eight aforementioned categories. In total, the dataset consists of 2,481 sentences with approximately 56,111 tokens. Each sentence contains up to 19 numerals.

3.1 Annotation Scheme and Taxonomy

For the deeper context extraction of numerals, the first challenge is to define the entity types. We begin by focusing on the questions with numerals in the SPIDER dataset [19]. We identify the numerals and look for those containing hidden information. In particular, we focus on numerals that do not have a token to describe what they are. For example, in the sentence "This year, I am 25", the numeral 25 denotes an age without any token as its description. This particular case can cause difficulty in recognition when a numeral's description or unit is lacking.

We propose eight classes in total for numeral classification. Below we summarize the annotation guidelines for the eight classes.

AGE is the age of anything such as people, animals, or plants, as well as buildings or places such as monuments, parks, or schools.
POPULATION is the number of inhabitants or capacity for inhabitants in a specific area—for example, a country's population, number of stadium seats, or number of enrolled students.
YEAR is a year in any format, e.g., in a 4-digit format such as 2021 or a 2-digit format such as 95.
DATE/MONTH is a specific month or date such as Sunday **24**th, the **8**th month, or **4**th of July.
LENGTH/HEIGHT is a measured size in two-dimensional space or time duration such as travel distance, human height, and running duration.
MONEY is any numeral related to money, such as a purchase amount, salary, or account balance.
WEIGHT/VOLUME is the measured weight or volume of anything such as a pet's weight, parcel's weight, or bottle's volume.
GENERIC is a broad category. A numeral that does not fall into any other category is considered to be part of the Generic type—for example, postcode, ID, or phone number.

In our annotation, we define a numeral entity as a token that contains only the numeral in the sentence without any characters or tokens describing its unit. However, we allow the following symbols in the numeral token to be annotated: "." (for floating-point numbers only), "-" (except when used to denote a range of values), and "/," as these symbols can exist in between numbers to connect multiple numeral groups to one entity, e.g., 3.14, 2021-01-01, and 2021/01/01.

3.2 Data Collection

The NumER dataset consists of documents gathered from four primary sources as follows.

- SPIDER [19] and SParC [20]: These text-to-SQL datasets consist of questions in formal and informal writing. Questions using numerals were collected.
- Wikidata [14]: This is an open data knowledge graph hosted by the Wikimedia Foundation. We queried the data using numeral properties both selectively and randomly. Sentences from the corresponding Wikipedia summaries describing the values of the selected properties were extracted.
- Epicurious[4]: A cooking recipe dataset. We extracted numeral-including sentences from the recipe instructions.
- News Category[5]: A dataset including news headlines from 2012 to 2018 obtained from HuffPost. Numeral-including sentences were extracted from the headlines and their summaries.

Annotation Process The dataset was annotated by Amazon Mechanical Turk (MTurk) workers. Every worker had to pass our qualification test to test their understanding of the taxonomy before working. The qualification test consisted of four questions about several numeral entities that could result in different decisions depending on whether a respondent fully understood the defined taxonomy. For example, following our taxonomy, the numeral "300" in the sentence "Find a theatre with a capacity above 300 seats" is a Population entity because 300 is the number of humans that can fit in such a theatre. From a different point of view, the annotator might think 300 is the *volume* of a theatre.

We assigned three different MTurk workers to annotate each numeral. We formulated the annotation task as a classification task to reduce the difficulty for both the workers and the implementation. We extracted numerals using a heuristic method. The token is considered to be a numeral when one comma and one period are replaced, and only digits are left. For the token that contains hyphen or slash symbols, if the token can be parsed as a date or time, we considered it as a single token. Otherwise, we split the token using those symbols. Then, we provided the extracted numeral and source sentence to the worker and asked them to categorize the given numeral.

[4] `https://www.kaggle.com/hugodarwood/epirecipes`
[5] `https://www.kaggle.com/rmisra/news-category-dataset`

Annotation Agreement After the annotation process was finished, 74% of numerals yielded consistent annotation results by receiving the same decision from all three annotators. 25% of numerals had a majority (two) decision from two annotators. Only 1% of numerals had a split decision resulting from differences in the annotators' decisions.

The Kappa score between every two annotators was 76.6%, 76.7%, and 76.8%, considered an "almost perfect" agreement [9].

Annotation Conflict The annotation conflicts were solved manually after calculating the inter-annotator agreement. For the majority-decision entities, we considered the majority decision as the correct entity type by default. Some sentences included both a majority-decision entity and a split-decision entity. We manually conducted further investigation to choose the final annotations for the split-decision entities to resolve conflicts.

3.3 Data Analysis

We split our data into training, development, and test sets by 70%, 10%, and 20%, respectively. We randomly split the data while maintaining the ratio between train/dev/test set in each class as close as possible to the ideal ratio. The general statistics of the dataset are presented in Table 1. Table 2 shows the data distribution in our dataset. The total number of entities is 5,477. Each class contains at least 300 entities. Year is the majority class with 1,580 entities. In contrast, Population is the minority class with 324 entities in total.

Each of our sentences contains at least one numeral and up to 19 numerals. 30% of our sentences contain at least three numerals in the sentence . Furthermore, 63% of the numeral tokens in our dataset belong to multiple classes and thus require disambiguation. For example, the numeral "20" can belong to the Age class, as in "age of 20", and the Length/height class, as in "20 cm.". These conditions can help provide a more complex situation for the model to tackle. In addition, more than half of our numeral entities are numerals without a unit token.

Table 1: NumER dataset statistics.

	Train	**Dev**	**Test**
Number of sentences	1,737	248	496
Number of token	39,573	5,457	11,081
Number of entities	3,825	537	1,115
type-token ratio	14%	30%	23%

3.4 Comparison to Other NER Datasets

As our dataset is focused on numeral tokens, there are only a few other datasets with the same focus. For example, Mandhan et al. [12] focused on numerals in a clinical text, such as those denoting blood pressure, temperature, pulse,

Table 2: Distribution of the numeral entities in the dataset.

Entity Type	**Train**	**Dev**	**Test**	**Total**
Age	329	47	94	470
Population	230	34	60	324
Year	1,139	141	300	1,580
Date/Month	647	81	177	905
Length/Height	536	98	206	840
Money	275	48	81	404
Weight/Volume	245	32	63	340
Generic	424	56	134	614
Total	3,825	537	1,115	5,477

heart rate, and drug dosage. Another example is NTCIR-14 FinNum [2], which focuses on the financial domain with informal documents gathered from Twitter. In contrast to these works, our dataset is focused on more general sentences in daily life.

4 Experimental Setup

This section describes the models used for benchmarking with our dataset. We benchmark two categories of models, including an off-the-shelf model and a pre-trained model that was fine-tuned on our training set. In every model, we configure the tokenizer to tokenize our focused numeral as one token. We also create another training set with a data augmentation technique to provide additional training data and improve the results.

4.1 Model Settings

spaCy 2.3.5 [6] The spaCy NER model uses deep convolutional neural network and transition-based named entity parsing. We use spaCy NER using a blank English model. We train the model from scratch with our training set for the maximum of 30 epochs with a learning rate of 0.001 until there is no improvement for three epochs.

BERT [3] We used BERT base models with both cased and uncased context, and fine-tuned the model on the NumER training set for three epochs with a learning rate of $5 \cdot 10^{-5}$ and a batch size of 32.

BiLSTM-CRF [5] We used the BiLSTM-CRF model with Glove embedding [11] and trained for 15 epochs maximum with a learning rate of 0.001, dropout of 0.5, and batch size of 20 until there was no improvement for three epochs.

[6] `https://v2.spacy.io/`

4.2 Data Augmentation

In our collected data, Population, Money, and Weight/Volume types can be considered minor classes because of the limited number of training data. To deal with the data sparsity, we create additional training data using the contextual augmentation technique [7] using the BERT language model. We randomly replace tokens in the sentence, including both words and numerals, with other suitable words predicted using BERT based on the original word's surrounding context. Thus, we keep the original label sequence unchanged.

Using the generated sentences, we created a new augmented dataset including the original sentences. In the augmentation, we replace randomly one to five tokens per sentence. For each original sentence, we generate five new sentences. In Table 3 we show examples of two source sentences and their augmented sentences.

Table 3: Example of data augmentation including two original sentences and three examples of augmented sentences for each.

Original Sentence	How many players have a weight greater than 220 or height shorter than 75?
Augmented	How can players have head weight greater than 220 its height shorter than 85? How would players have any breadth greater than 60 or height shorter than 75? How many players has their weight increased than 220 or height smaller than 75?
Original Sentence	It was built in 1974 to a height of 123 metres.
Augmented	Bridge was built in 1922 to a height spanning 123 metres. It was built about 1974 to a height over 123 metres. It was built since 1974 to a heights of 123 metre.

Because this method relies on randomization in choosing the token to replace, there is a chance that the predicted token from the language model may change the context of the sentence. This change can affect the entity type of a numeral token when its neighbour token is changed. To ensure the numeral's validity and type in the augmented sentences, we performed a manual check for every sentence that changed in the tokens near the annotated numeral token.

As a result of the augmentation process, 6,146 sentences are generated by contextual augmentation in addition to the original 1,737 training sentences, yielding a total of 7,883 sentences in the augmented training set. The development and testing set consists of the original 248 and 496 sentences, respectively.

5 Results

The models are evaluated on our test set to obtain entity-level precision, recall, and F1-score per class. The results obtained using an off-the-shelf model and trained models are reported in Table 4. We determine that all trained/fine-tuned models work better than the off-the-shelf tools. Every model achieved 100% F1-score in the span detection task. Overall, using the BERT model, we can achieve

an F1-score of 95.2%. Date/month had the best results while Population had the worst. Using BiLSTM-CRF, we can reach an F1-score of 88.5%.

The overall best-performing model is BERT-cased fine-tuned on the NumER augmented training set. And every model trained on the augmented dataset performed better than those trained on the non-augmented dataset. It is encouraging that the BERT-based models are also able to capture the context of the numeral tokens. Table 5 describe each class's score of the best models.

Table 4: Overall results of baseline models in the NumER dataset.

Architecture	**Training Set**	**Precision**	**Recall**	**F1-Score**
BERT-cased	Augmented	0.953	0.952	0.952
BERT-cased	Non-augmented	0.943	0.936	0.938
BERT-uncased	Augmented	0.948	0.946	0.947
BERT-uncased	Non-augmented	0.947	0.947	0.946
BiLSTM-CRF	Augmented	0.886	0.884	0.885
BiLSTM-CRF	Non-augmented	0.865	0.862	0.863
spaCy	Augmented	0.856	0.855	0.856
spaCy	Non-augmented	0.820	0.818	0.818

Table 5: The results of the BERT-uncased model without data augmentation (UC-NOAUG), the BERT-cased model with data augmentation (C-AUG), and the SpaCy model with data augmentation (SpaCy-AUG). (P = Precision; R = Recall; F1 = F1-Score)

	UC-NOAUG			**C-AUG**			**SpaCy-AUG**		
Entity Type	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Age	0.979	0.979	0.979	0.978	0.947	0.962	0.950	0.809	0.874
Population	0.791	0.883	0.835	0.794	0.833	0.813	0.533	0.706	0.608
Year	1.000	0.980	0.990	1.000	0.983	0.992	0.952	0.986	0.979
Date/Month	0.967	1.000	0.983	1.000	1.000	1.000	0.975	0.975	0.975
Length/Height	0.965	0.951	0.958	0.943	0.976	0.959	0.833	0.816	0.825
Money	0.920	0.988	0.952	0.952	0.988	0.970	0.950	0.792	0.864
Weight/Volume	0.846	0.873	0.859	0.814	0.905	0.857	0.533	0.813	0.658
Generic	0.885	0.812	0.847	0.917	0.835	0.874	0.854	0.625	0.722
Total	0.947	0.947	0.946	0.953	0.952	0.952	0.856	0.855	0.856

The results indicate that our augmentation technique helps the models to perform slightly better than they do using only the original data. For the models trained with non-augmented data, the worst-performing class is Population with an F1-score of 73.4%. After using the augmented data for training, the F1-score improves by almost 8% in the BERT-uncased model, which is the most significant improvement per class.

6 Application to Text-to-SQL Task

To demonstrate the benefit of the NumER dataset, we experimented on text-to-SQL tasks by incorporating the information from NumER into an existing

text-to-SQL model. Such a process usually requires schema linking, a process to match the candidate value token in the question to its associated column. The numeral entity type information can benefit the schema linking process in the model. Namely, given the numeral entity type information, the model can perform the schema linking even when there is no overlap token between the query and candidate column names/values. Note that this is difficult for existing models that are based on surface-level string matching.

6.1 Model

We modified the IRNET model [4], a text-to-SQL model trained on the SPIDER dataset. IRNET is based on encoder-decoder architecture with a memory augmented pointer network. The input embedding for the natural language schema encoders is concatenated with additional information from NumER. Furthermore, the schema linking process is enhanced to consider our numeral entity types. The three modified components are described below.

First, we modified the question token type embedding which describes the referred schema and SQL command-related component in each token, including table, column, aggregated function, comparative word, superlative word, and numeral. We extended the embedding by adding eight more features to represent each NumER entity type using one-hot encoding.

Second, the column type embedding, which is used to keep track of which column is mentioned in the input question, is modified. We extended the embedding with eight more features in the same way as to question token type using the information from our manually annotated type of each column.

Finally, we modified the schema linking process in the preprocessing step. We performed the original schema linking process first. Then the type of each numeral token is recognized using the NumER model. We map the detected numeral type to the column with the same type. If there are multiple candidate columns, the column in the table with the matching name in question tokens is selected. If the mapped column was not detected in the original process, we add the token indicating the mapped column name in front of the numeral.

6.2 Result

We trained our modified version of IRNET model on the SPIDER training data and evaluated it using the SPIDER development dataset on the "exact set match without values" setting. We achieved 58.4% accuracy, compared to the vanilla IRNET model with 53.2% accuracy. As a result, the model benefits from the NumER model's information and has a performance improvement of 5.2%.

7 Conclusion

In this work, we present NumER, a fine-grained numeral entity recognition dataset that successfully classified numerals in a more generic domain than that

of the typical NER dataset. The data consisted of non-specific-domain sentences from several sources. The collected sentences were in the form of articles, question, titles, and instructions. We conducted experiments by training models on our dataset and benchmarked using well-known models.

According to the results, the models can successfully capture the semantics of the numeral token. This shows that (I) our method can be used to extract information from numerals in the sentence, and (II) the numeral classification in a more generic domain is also possible and not limited to just a specific domain. In the future, we can extend our proposed taxonomy to more classes or adapt to match well-known ontology. We will also apply our model to current NLP challenges involving numerals to improve the result of target tasks and extend our data's usefulness.

References

1. Azzi, A.A., Bouamor, H.: Fortia1@ the NTCIR-14 FinNum task: enriched sequence labeling for numeral classification. In: Proceedings of the 14th NTCIR Conference on Evaluation of Information Access Technologies. pp. 526–538 (2019)
2. Chen, C.C., Huang, H.H., Takamura, H., Chen, H.H.: Overview of the NTCIR-14 FinNum task: Fine-grained numeral understanding in financial social media data. In: Proceedings of the 14th NTCIR Conference on Evaluation of Information Access Technologies. pp. 19–27 (2019)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423
4. Guo, J., Zhan, Z., Gao, Y., Xiao, Y., Lou, J.G., Liu, T., Zhang, D.: Towards complex text-to-SQL in cross-domain database with intermediate representation. In: Proceeding of the 57th Annual Meeting of the Association for Computational Linguistics (ACL). pp. 4524–4535. Association for Computational Linguistics (01 2019). https://doi.org/10.18653/v1/P19-1444
5. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. CoRR **abs/1508.01991** (2015)
6. Jiang, M.T.J., Chen, Y.K., Wu, S.H.: CYUT at the NTCIR-15 finnum-2 task: Tokenization and fine-tuning techniques for numeral attachment in financial tweets. In: Proceedings of the 15th NTCIR Conference on Evaluation of Information Access Technologies. pp. 92–96 (2020)
7. Kobayashi, S.: Contextual augmentation: Data augmentation by words with paradigmatic relations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). pp. 452–457. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018). https://doi.org/10.18653/v1/N18-2072
8. Min, K., MacDonell, S., Moon, Y.J.: Heuristic and rule-based knowledge acquisition: Classification of numeral strings in text. In: Hoffmann, A., Kang, B.h., Richards, D., Tsumoto, S. (eds.) Advances in Knowledge Acquisition and

Management. pp. 40–50. Springer Berlin Heidelberg, Berlin, Heidelberg (2006). https://doi.org/10.1007/11961239_4
9. Munoz, S., Bangdiwala, S.: Interpretation of Kappa and b statistics measures of agreement. Journal of Applied Statistics **24**, 105–112 (02 1997). https://doi.org/10.1080/02664769723918
10. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticæ Investigationes **30**(1), 3–26 (2007). https://doi.org/10.1075/li.30.1.03nad
11. Pennington, J., Socher, R., Manning, C.D.: GloVe: Global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543 (2014). https://doi.org/10.3115/v1/D14-1162
12. R., S.P., Mandhan, S., Niwa, Y.: Numerical atrribute extraction from clinical texts. CoRR **abs/1602.00269** (2016). https://doi.org/10.13140/RG.2.1.4763.3365
13. Tjong Kim Sang, E.F., De Meulder, F.: Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003. pp. 142–147 (2003), https://www.aclweb.org/anthology/W03-0419
14. Vrandečić, D., Krötzsch, M.: Wikidata: A free collaborative knowledgebase. Commun. ACM **57**(10), 78–85 (Sep 2014). https://doi.org/10.1145/2629489
15. Weischedel, Ralph, Palmer, Martha, Marcus, Mitchell, Hovy, Eduard, Pradhan, Sameer, Ramshaw, Lance, Xue, Nianwen, Taylor, Ann, Kaufman, Jeff, Franchini, Michelle, El-Bachouti, Mohammed, Belvin, Robert, Houston, Ann: OntoNotes release 5.0 (2013). https://doi.org/10.35111/XMHB-2B84
16. Wu, Q., Wang, G., Zhu, Y., Liu, H., Karlsson, B.: DeepMRT at the NTCIR-14 finnum task: A hybrid neural model for numeral type classification in financial tweets. In: Proceedings of the 14th NTCIR Conference on Evaluation of Information Access Technologies. pp. 585–595 (2019)
17. Yadav, V., Bethard, S.: A survey on recent advances in named entity recognition from deep learning models. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 2145–2158. Association for Computational Linguistics, Santa Fe, New Mexico, USA (Aug 2018), https://www.aclweb.org/anthology/C18-1182
18. Yadav, V., Bethard, S.: A survey on recent advances in named entity recognition from deep learning models. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 2145–2158. Association for Computational Linguistics, Santa Fe, New Mexico, USA (Aug 2018), https://www.aclweb.org/anthology/C18-1182
19. Yu, T., Zhang, R., Yang, K., Yasunaga, M., Wang, D., Li, Z., Ma, J., Li, I., Yao, Q., Roman, S., Zhang, Z., Radev, D.: Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 3911–3921. Association for Computational Linguistics, Brussels, Belgium (Oct-Nov 2018). https://doi.org/10.18653/v1/D18-1425
20. Yu, T., Zhang, R., Yasunaga, M., Tan, Y.C., Lin, X.V., Li, S., Er, H., Li, I., Pang, B., Chen, T., Ji, E., Dixit, S., Proctor, D., Shim, S., Kraft, J., Zhang, V., Xiong, C., Socher, R., Radev, D.: SParC: Cross-domain semantic parsing in context. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 4511–4523. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1443