A Pragmatic Approach to Information Extraction for Systematic Review

Adyasha Maharana, Arpit Tandon, Eric Wimberley, Mihir Shah,
Ruchir Shah, Brian E. Howard
Sciome LLC
{adyasha.maharana,arpit.tandon,charles.wimberley,mihir.shah,
ruchir.shah,brian.howard}@sciome.com

Abstract

This report describes the performance of various named entity recognition methods for extraction of data elements for systematic review in the TAC SRIE 2018 challenge. The approach uses a combination of conditional random fields and recurrent neural nets with multi-task learning and achieved 49.76% overall F1-score on the test dataset. Our system was placed second in the challenge. We also demonstrate the efficacy of sentence classification as an alternative way to aid manual extraction from scientific text.

1 Introduction

Biomedical text is one of the most widely studied application domains in information extraction. Years of ongoing research has led to the development of effective, scalable techniques for critical tasks like gene and protein entity recognition (Settles, 2005), adverse drug event detection (Sarker and Gonzalez, 2015), protein interaction (He et al., 2009), etc. Much of the research impetus for these efforts is provided by the release of publicly available corpuses like BioCreative tasks (Dogan et al., 2017; Hirschman et al., 2005; Li et al., 2016), TAC ADE (Demner-Fushman et al., 2018), GENIA (Kim et al., 2003) and i2b2 (Uzuner and Stubbs, 2015) among others. Tools developed using these methods and datasets are extensively used by scientists to aid literature review processes and expedite retrieval of structured knowledge.

Systematic review is one such intensive process of collection, analysis and summarization of empirical evidence that is used to reliably answer a given research question. A critical and time-consuming process that must occur during systematic review is the extraction of relevant qualitative and quantitative raw data from the free text of scientific documents. The resulting data is used to answer the review's research question(s) and as inputs to various forms of meta-analysis. The specific data types extracted differ among disciplines, but within a given scientific domain, certain data points are extracted repeatedly for each new review that is conducted. Given the extremely laborious, time intensive, and repetitive nature of this data extraction step, systematic review practitioners have long been interested in the possibility of automated or semi-automated information extraction from scientific documents. Except for a few tasks on PICO elements (Kim et al., 2011; Nye et al., 2018), information extraction for systematic reviews has remained largely unexplored due to the lack of datasets. For this reason, organizers from NIEHS and EPA released an annotated corpus and held the TAC SRIE 2018 information extraction challenge to assess and advance the state-of-the art for information extraction in the context of systematic review.

This report summarizes our contributions to the challenge and the methods we have evaluated for extraction of SRIE Task entities.

2 Dataset

The TAC SRIE training dataset consists of 100 annotated 'Materials and Methods' sections from research articles. The dataset is annotated with 24 entity classes, which are further grouped into 4 major categories: *Exposure*, *Animal Group*, *Dose Group* and *Endpoint*. There are 8,353 sentences and the average sentence length is 24. Besides multiple paragraphs, the 'Materials and Methods' sections also contain sub-headings which are delineated using newline characters. The most frequently occurring entity classes are *Endpoint*, *TestArticle*, *Species* and *GroupName* with 6,533, 1,961, 1,624 and 1,119 entities respectively (see Table A1 in the Appendix.) The least frequently

occurring entity classes are *TestArticleVerification*, *TestArticlePurity*, *TimeAtLastDose*, *TimeAtFirstDose*, *CellLine* and *SampleSize*, each with an average count of <1 per document. There are a significant number of occurrences of inter-class as well as intra-class overlap in annotations. In addition, nearly 14.3% of the 15,267 entities span over discontinuous fragments of tokens within the document.

3 Methods

Due to the varying frequencies of the diverse entity categories in our dataset, we adopted different approaches for different entities contingent on the amount of labeled data available. For high-frequency entity classes like Endpoint and TestArticle, we experimented with deep-learning based named entity recognition methods. Deep Learning based methods are known to generalize better for biomedical text (Wu et al., 2017; Maharana and Yetisgen, 2017; Sachan et al., 2017). For entity classes having an intermediate amount of training data, we used Conditional Random Fields (CRFs) (Lafferty and Mccallum, 2001). Lastly, for those entity classes having only a small number of labeled training instances, we used rules and regular expressions. See Table 2 in the Appendix for a complete breakdown of the methods employed for each Task I entity class.

Several entity classes are closely related to one another in terms of semantics, relative position within the text, and shared, overlapping words. Some of these similarities can be leveraged to build models that jointly predict more than one category. To better understand these similarities, we computed co-occurrence counts for each entity class with respect to all other entity classes. Some of the conclusions that stood out from this analysis were:

- Nearly 80% of Strain and 50% of Sex entities co-occur with Species entities within the same sentence.
- Dose and DoseDuration entities are almost always accompanied by their respective DoseUnits and DoseDurationUnits entities. DoseFrequency co-occurs with 27.8% of the DoseDuration entities.
- DoseRoute entities co-occur with nearly half of the Dose entities and only a quarter of TestArticle entities.
- A third of EndpointUnitOfMeasure entities have overlapping spans with Endpoint entities.

We used some of these insights to group entity classes together in joint prediction models.

3.1 Preprocessing

The documents in the training set were separated into individual sentences using the Punkt Sentence Tokenizer. These tokenized sentences were then passed through the GENIA tagger (Tsuruoka et al., 2005) to augment the given text with a comprehensive output containing tokenized words and their respective lemmas, part-of-speech (POS) tags, chunk tags and GENIA named-entity tags (Protein, DNA, RNA, Cell Line and Cell Type). These entity annotations were then converted into the BIO (Begin-, Inside-, Outside-Entity) format for downstream processing. Those entities which consist of discontinuous spans were broken into individual continuous entities for the sake of modelling.

3.2 Deep Learning for Named Entity Recognition

Recurrent Neural Networks (RNN) are considered state-of-art algorithms for most NLP tasks, though they have been outperformed on some tasks by recent Transformer architectures (Vaswani et al., 2017). In order to realize these performance gains, RNN-based models must be initialized with pre-trained word embeddings and then trained using a sizable task-specific dataset. They are also often augmented with a sequence optimization layer for sequence labelling tasks like named entity recognition and part-of-speech tagging. Recent developments in Natural Language Processing have shown that pre-trained language models can be used with task-specific RNNs to further improve performance via transfer learning (Howard and Ruder, 2018).

3.3 Embeddings

To initialize our deep learning models, we used pre-trained biomedical word2vec word embeddings (dimension = 200) extracted from the PubMed and PMC databases, encompassing in total over five billion words (Pyysalo et al., 2012).

3.4 Bidirectional RNN with Sequence Optimization

The NER architecture introduced in (Lample et al.,) has been established as a strong baseline for several open-domain and biomedical entity recognition datasets. It consists of five components – a character embedding layer, a bidirectional character LSTM, a word embedding layer, a bidirectional word LSTM, and an integrated probabilistic CRF model. It is trained end-to-end by optimizing the negative log-likelihood loss between predicted and target labels. We used this BiLSTM-CRF architecture as a baseline for our experiments with *Endpoint*, *TestArticle* and *DoseGroup* entity classes.

3.5 Multi-task Sequence Labelling Models

Multi-task learning is a well-known technique used in machine learning to improve generalization of models by sharing representation with one or more related tasks (Caruana, 1997). In simpler words, the model is forced to first learn easy tasks and then use those skills to master more complex tasks. Usually, this involves optimizing more than one loss function simultaneously while training the model. We have experimented with two auxiliary tasks (in addition to entity recognition) in the multi-task setting – language modelling and sentence classification.

NER + Character-level language model (LM-LSTM-CRF): We used the task-aware neural language model proposed in Liu et al. (2017) to jointly train a character-level language model and an entity recognizer. Character representations from bi-directional LSTM layers were mapped onto two different semantic spaces via residual layers for language modelling and sequence labelling. The language modelling objective was combined with the NER objective using a learnable weight parameter.

NER + Sentence Classification (S-LSTM-CRF): The final word representation from the bi-directional LSTM layer in a BiLSTM-CRF model was mapped onto a different semantic space using a fully-connected layer and passed into a soft-max layer for binary sentence level classification. Sentences are classified into 'contains one or more entities' or 'doesn't contain any entity' category; the entity class depends on the target NER task. The binary cross-entropy classification objective is combined with negative log-likelihood objective using a fixed weight parameter.

3.6 Training and Hyper-parameters

In general, the NER datasets for individual categories were very unbalanced – i.e. the number of sentences containing no entity (zero-entity sentences) is at least three times the number of sentences which have an entity. For example, only 2,353 sentences feature an Endpoint entity out of 8,353 sentences in the training dataset. The number drops even lower to 1,278 and 535 for TestArticle and GroupName, respectively. To counter the unbalanced-ness, we adopt undersampling of zero-entity sentences during training. For every training epoch, a new, randomly picked subset of zero-entity sentences was used for training while the rest were discarded. This ensures that the model gets to see all data points during training.

Features in Conditional Random Field Model	
• Word (n-grams) • Boolean feature for title case • POS tag (Genia) • Lowercase of word • Boolean feature for numeric characters • Chunk tag (Genia)	• Lemma (Genia) • Character n-grams (prefix and suffixes) • Length of word • Word shape (X, x, 0, special-char) • Boolean feature for uppercase

Table 1: Common features in Conditional Random Field models for SRIE dataset

Target Entities	**Additional features**
Strain, GroupSize, SampleSize, GroupName	Output from *CellLine*, *Strain*, *Sex* and *Species* regular expression models were used as features
EndpointUnitOfMeasure	Endpoint entities were used as features
TimeUnits	Augmented with lexicon of common time units

Table 2: List of CRF models and their respective target entities

Due to time constraints, we did not perform extensive hyperparameter optimization for deep learning architectures. Apart from a few changes, we have used the recommended settings for BiLSTM-CRF and LM-LSTM-CRF networks (Lample et al., 2016; Liu et al., 2017). The dimensions for character LSTM and word LSTM representations are 50 and 200 respectively. Dropout rate of 0.5 was applied on word and

character embedding layers during training. We also experimented with the usage of POS tags and chunk tags as feature embeddings, however this did not lead to any significant improvement in performance. The Stochastic Gradient Descent (SGD) algorithm was used for training the networks and the learning rate was fixed after optimization via cross-validation experiments.

3.7 Rules & Regular Expression Models

For the *Sex*, *Species*, *Strain*, *Vehicle*, and *CellLine* entity classes, we developed models based on dictionaries, regular-expression rules and GENIA tags. Orthographic structure of the words and word position relative to other entity categories were the most important cues for identifying *CellLine* and *Strain* entities, and these are therefore incorporated into the various rule-based models created for this task.

3.8 Conditional Random Fields

We used conditional random fields (CRFs) to build models for those entity classes having less than 1,000 available data points in the training set, since neural networks are prone to overfitting with smaller datasets. All CRF models have a common set of basic features, and additional entity-specific features and hyperparameters. The window size for word and character n-gram features for all CRF models varies between 3-5. The set of basic textual features used in all CRF models can be found in Table 1.

In addition to feature engineering, we also adopted under-sampling of zero-entity sentences, joint modelling of entities (Table 2) and cascading with regular expression models to improve performance for certain entities. We used co-occurrence counts to group entities together that might benefit from joint prediction. For *EndpointUnitOfMeasure*, we also performed propagation of tagged entities throughout the document text.

3.9 Ensembling

To evaluate each of our statistical models, we use five-fold cross-validation: i.e. we built five separate models, trained and tested on different partitions of the training dataset. To get results for the TAC SRIE evaluation dataset, we ran each of the five models on the evaluation corpus and merged results using a voting-based ensemble.

3.10 Document Level Models

There are multiple cross-sentence dependencies between pairs of entity classes such as *Endpoint* and *EndpointUnitOfMeasure*, *TestArticle* and *Vehicle*, etc. Even within the same category, a statistical model can benefit from using the entire document as context for named entity recognition within a single sentence. To this end, we attempted to build a document-level CRF model that takes tagged entities from existing models and uses features from the entire document to re-tag entities within each sentence. However, we did not see any improvement in performance using this model, and therefore did not include this model in our submission. In future work, we will experiment with other methods for using the document as context.

3.11 Sentence Classification

We were curious if courser-grained, sentence-level annotations might result in potentially useful models for extraction. We trained sentence classifiers for the four major tag types, as defined in the SRIE Annotation Guidelines: Exposure, Animal Group, Dose Group and End Point. Each sentence was annotated as a positive example of the class if it contained an annotation for one of the corresponding sub-tags (e.g. “TestArticle” or “Vehicle” for the Exposure tag). We evaluated several approaches including traditional classifiers (logistic regression, SVM, random forest and naïve Bayes) based on bag-of-words, topic model and/or vector-embeddings feature sets. We also tested several artificial neural network architectures including several varieties of LSTM classifier models, including the multi-task ULMFit model from fast.ai. The ULMFit model uses a language model pretrained on a large text corpus to implement transfer learning in the context of text-classification (Howard and Ruder, 2018).

4 Results

Endpoint Model	Data	Recall	Precision	F1
LSTM-CRF (baseline)	Train	**0.744**	0.448	0.559
	Test	**0.643**	0.365	0.465
LM-LSTM-CRF	Train	0.721	0.552	**0.625**
	Test	0.599	0.476	**0.531**
S-LSTM-CRF	Train	0.586	**0.612**	0.599
	Test	0.534	**0.478**	0.505

Table 3: Comparative Analysis of various Endpoint models on SRIE training and evaluation corpus

Endpoint Model	Data	Recall	Precision	F1
LSTM-CRF (baseline)	Train	**0.744**	0.448	0.559
	Test	**0.643**	0.365	0.465

Table 4: Comparative Analysis of TestArticle model on SRIE training and evaluation corpus

4.1 Named Entity Recognition

We submitted models for 19 out of the 24 entity categories in Task 1 of the SRIE dataset. Table A2 contains the full set of results for each of these models on both the training and test datasets.

For Endpoint, we had results from three separate models: BiLSTM-CRF, LM-LSTM-CRF and S-LSTM-CRF. The best F1-score was obtained from the LM-LSTM-CRF multi-task model. However, none of the multi-task models could improve on the recall score of the baseline LSTM-CRF. The S-LSTM-CRF model had the best precision score. The complementary strengths of these three models could be potentially combined through stacked ensembling methods for a better performing Endpoint model. A similar trend in performance was also observed in the evaluation set, however, there was a nearly 10% absolute drop in the best performing model's F1-score.

We experimented with all the above deep learning architectures for predicting entity tags for TestArticle and with different undersampling rates. We also used the off-the-shelf ChemNER model to find chemical entities and feed them as input to the models as feature embedding. The baseline, BiLSTM-CRF, remained the best performing model for TestArticle despite several modifications in feature set and architecture. However, the performance suffered a sharp drop in the evaluation set, suggesting that the data distribution of TestArticle entities in evaluation set varies significantly from that of training set.

Among other noteworthy results, the F1-score of the *DoseFrequency* model increased by 10% for the evaluation set. The performance scores on training and evaluation data for rest of the *DoseGroup* entity classes were similar, except for *Dose* and *DoseUnits*. The CRF model for *EndpointUnitOfMeasure* depends on accurate input of Endpoint entities. Therefore, it isn't surprising to see that it also fares worse on the evaluation set. Most of the regular expression/rules/lexicon-based models performed worse on the evaluation set. In this case, the drop in performance could be attributed to overfitting by rules and limited lexicon coverage

4.2 Sentence Classification

The best performing model for the sentence classification task was the ULMFit model from fast.ai. The following table shows the results of 5-fold cross validation on the training dataset.

With respect to the other neural network models, the best performing model was the bi-directional LSTM with character level extraction, dropout for embeddings only, automatic learning rate adjustment, and use of "UNK" to replace rare words. F1-Scores for this approach were 3-5% lower for each tag (data not shown). Among the traditional machine learning classifiers, logistic regression with tf-idf weighted bag-of-words features provided the best performance.

5 Discussion & Conclusion

From our experiments, we can conclude that multi-task learning is beneficial for improving performance of Named Entity Recognition models on biomedical datasets. However, the auxiliary task needs to be carefully chosen to optimize for sensitivity or specificity. On this dataset, addition of a language modelling objective lead to optimization for both recall and precision, while the binary sentence classification objective optimized mostly for precision.

One of the possibilities that remained unexplored in our experiments is the use of better word embeddings to initialize the recurrent neural network architectures. Several NLP tasks have received substantial boost in performance from using deep contextualized word representations,

Label	Accuracy	Recall	Precision	F1
AnimalGroup	0.969	0.926	0.938	0.932
DoseGroup	0.938	0.647	0.851	0.735
Endpoint	0.885	0.805	0.813	0.809
Exposure	0.926	0.754	0.834	0.792

Table 5: Sentence classification performance

better known as ELMo embeddings . Initializing our multi-tasking networks with ELMo embeddings could lead to further improvement in tagging of Endpoint and TestArticle entities. More recently, Transformers, which are non-recurrence neural networks that depend solely on an attention mechanism, have outperformed state-of-the-art architectures in several NLP tasks. Bi-directional Transformers (BERT) can be fine-tuned to target tasks after being pre-trained on a large open domain corpus (Devlin et al., 2018). However, these are data-hungry networks and we run the risk of severely overfitting them on a corpus as small as the SRIE dataset. With the availability of more training data, it will become increasingly possible to leverage advanced deep learning methods.

Named Entity Recognition has been traditionally performed at the sentence level. For the most part, this works fine for open-domain datasets because entities like Person, Location, and Organization do not need contextual information from other parts of the document. The SRIE entity recognition dataset differs in this regard since tagging of TestArticle, Endpoint etc., in one part of the text is informed by cues from several other parts of the text. For example, AuNP is a test article entity in the sentence "AuNP aerosol generation and inhalation were performed as previously described for titanium dioxide NP [27] and gold NP [28]", but it wouldn't be immediately clear to the reader without context from a previous sentence, "we studied (i) AuNP distribution in lung tissue and (ii) AuNP uptake by surface macrophages, at the individual particle level". If annotators had to tag the SRIE dataset by looking at only one sentence at a time, the resulting annotations might not cover more than 60-70% of the annotations currently present in the dataset. Statistical methods will quickly hit this upper cap even with the use of more sophisticated embeddings and NER architectures. We attempted to solve this problem by a) propagating all predicted TestArticle entities throughout the text and b) propagating only high confidence TestArticle entities. The corresponding sentence classification score is used as a confidence score for this analysis. We observed up to 1.5% improvement in F-score using this method (Table 6). Observed gains are also likely reduced by spurious False Positives, caused by inconsistent annotations in the dataset. Nevertheless, the gains aren't large enough to push model performance to the level of humans. The bottleneck still lies in framing of this task at the level of the sentence. We propose to reframe this task as a document level sequence labelling task so as to allow the model to leverage information from elsewhere in the text. To this end, we engineered document-level features for a CRF model that makes predictions for a sentence, but we have not yet been successful. However, this approach holds promise and there is much to be explored in terms of methods to make this approach feasible (Luo et al., 2018).

	TP	FP	FN	F1
Baseline	1017	928	857	53.26
Score Threshold = 2.00	1157	1235	717	54.23
Score Threshold = 2.50	1158	1199	716	54.74
Score Threshold = 2.75	1138	1172	736	54.40
Score Threshold = 3.00	1127	1149	747	54.31

Table 6: Performance after propagation of predicted TestArticle entities at different thresholds

The results shown in Table 5 suggest that annotations performed at the sentence level may exhibit improved performance over the finer-grained entities required by the challenge. For example, our best model was able to identify Endpoint-related sentences (including sentences tagged with the *Endpoint*, *EndpointUnitOfMeasure*, and/or *TimeEndpointAssessed* entities) with an F1 score of .809 (using cross-validation on the training data), as compared to an F1 score of .599 for the Endpoint entity. This is to be expected since sentence-level classification does not require exact detection of entity boundaries and has the advantage of pooling training data across multiple similar categories. Nevertheless, in a practical data extraction scenario, sentence-level methods (and other approximate extraction tasks), especially when incorporated into user-friendly software applications, may still be useful for implementing computer-assisted extraction applications in a

semi-automated way. On the other hand, at this time, completely automated data extraction is most likely not feasible for all possible entities.

6 Conclusion

Although much work remains to be done to successfully integrate automated or semi-automated data extraction into a typical Systematic Review workflow, the 2018 SRIE TAC is a bold step in the right direction. Furthermore, Sciome is committed to working with the community to translate and integrate the latest information extraction models and methods into a practical, extensible data extraction platform that will continue to improve along with the rapidly advancing state-of-the-art.

References

Rich Caruana. 1997. Multitask Learning. *Machine Learning*.

Dina Demner-Fushman, Sonya E Shooshan, Laritza Rodriguez, Alan R Aronson, Francois Lang, Willie Rogers, Kirk Roberts, and Joseph Tonning. 2018. A dataset of 200 structured product labels annotated for adverse drug reactions. *Scientific data*, 5:180001.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Rezarta Islamaj Dogan, Andrew Chatr-aryamontri, Sun Kim, Chih-Hsuan Wei, Yifan Peng, Donald Comeau, and Zhiyong Lu. 2017. BioCreative VI Precision Medicine Track: creating a training corpus for mining protein-protein interactions affected by mutations. In *BioNLP 2017*, pages 171–175.

Min He, Yi Wang, and Wei Li. 2009. PPI finder: a mining tool for human protein-protein interactions. *PloS one*, 4(2):e4554.

Lynette Hirschman, Alexander Yeh, Christian Blaschke, and Alfonso Valencia. 2005. Overview of BioCreAtIvE: critical assessment of information extraction for biology.

Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification.

J-D Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. GENIA corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182.

Su Nam Kim, David Martinez, Lawrence Cavedon, and Lars Yencken. 2011. Automatic classification of sentences to support evidence based medicine. In *BMC bioinformatics*, volume 12, page S5. BioMed Central.

John Lafferty and Andrew Mccallum. 2001. Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data. , 2001(June):282–289.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer 2016. Neural Architectures for Named Entity Recognition.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2016. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database*, 2016.

Liyuan Liu, Jingbo Shang, Xiang Ren, Frank F Xu, Huan Gui, Jian Peng, and Jiawei Han. 2017. Empower Sequence Labeling with Task-Aware Neural Language Model.

Ling Luo, Zhihao Yang, Pei Yang, Yin Zhang, Lei Wang, Hongfei Lin, and Jian Wang. 2018. An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition. *Bioinformatics*,

34(8):1381–1388, April.

Adyasha Maharana and Meliha Yetisgen. 2017. Clinical Event Detection with Hybrid Neural Architecture. In *BioNLP*, pages 351–355.

Benjamin Nye, Junyi Jessy Li, Roma Patel, Yinfei Yang, Iain J Marshall, Ani Nenkova, and Byron C Wallace. 2018. A corpus with multi-level annotations of patients, interventions and outcomes to support language processing for medical literature. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2018, page 197. NIH Public Access.

Sampo Pyysalo, Filip Ginter, Hans Moen, Tapio Salakoski, and Sophia Ananiadou. 2012. Distributional Semantics Resources for Biomedical Text Processing.

Devendra Singh Sachan, Pengtao Xie, Mrinmaya Sachan, and Eric P Xing. 2017. Effective Use of Bidirectional Language Modeling for Transfer Learning in Biomedical Named Entity Recognition. *arXiv preprint arXiv:1711.07908*.

Abeed Sarker and Graciela Gonzalez. 2015. Portable automatic text classification for adverse drug reaction detection via multi-corpus training. *Journal of biomedical informatics*, 53:196–207.

Burr Settles. 2005. ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text. *Bioinformatics*, 21(14):3191–3192.

Yoshimasa Tsuruoka, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, John McNaught, Sophia Ananiadou, and Jun'ichi Tsujii. 2005. Developing a Robust Part-of-Speech Tagger for Biomedical Text. In *Advances in Informatics - 10th Panhellenic Conference on Informatics, LNCS 3746*, pages 382–392.

Özlem Uzuner and Amber Stubbs. 2015. Practical applications for natural language processing in clinical research: The 2014 i2b2/UTHealth shared tasks. *Journal of biomedical informatics*, 58(Suppl):S1.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. (Nips).

Yonghui Wu, Min Jiang, Jun Xu, Degui Zhi, and Hua Xu. 2017. Clinical Named Entity Recognition Using Deep Learning Models. *AMIA ... Annual Symposium proceedings. AMIA Symposium*, 2017:1812–1819.

A Appendices

Entity Class	No. of Mentions in Training Set	No. of Mentions in Test Set
Exposure		
TestArticle	1961	2312
Vehicle	446	371
TestArticlePurity	28	19
TestArticleVerification	7	2
Animal Group		
GroupName	1119	1225
GroupSize	394	503
SampleSize	45	74
Species	1624	1639
Strain	379	355
Sex	612	608
CellLine	40	110
Dose Group		
Dose	660	611
DoseUnits	498	444
DoseFrequency	101	106
DoseDuration	217	190
DoseDurationUnits	204	177
DoseRoute	589	543
TimeAtDose	133	61
TimeUnits	614	754
TimeAtFirstDose	53	73

TimeAtLastDose	23	46
Endpoint		
Endpoint	6533	5637
EndpointUnitOfMeasure	727	716
TimeEndpointAssessed	751	993

Table A1: Occurrence counts of all entity classes for SRIE training and evaluation corpus

Entity Class	**Model Type**	**F-Score (T)**	**F-Score (E)**
Sex	Rules/Regex	0.94	0.76
Species	Rules/Regex	0.95	0.81
Vehicle	Rules/Regex	0.56	0.34
CellLine	Rules/Regex	0.75	0.45
TimeAtDose	Rules/Regex	0.36	0.11
Strain	Rules/Regex	0.66	--
	CRF	0.74	0.55
GroupName	CRF	0.62	--
GroupSize	CRF	0.71	0.62
SampleSize	CRF	0.31	0.27
EndpointUnitOfMeasure	CRF	0.40	0.24
TimeUnits	CRF	0.62	0.47
Endpoint	LSTM-CRF	0.56	0.46
	LM-LSTM-CRF	0.63	0.53
	S-LSTM-CRF	0.60	0.50
TestArticle	LSTM-CRF	0.56	0.38
Dose	LSTM-CRF	0.70	0.55
DoseUnits	LSTM-CRF	0.72	0.51
DoseDuration	LSTM-CRF	0.47	0.46
DoseDurationUnits	LSTM-CRF	0.48	0.46
Dose Frequency	LSTM-CRF	0.42	0.51
DoseRoute	LSTM-CRF	0.63	0.62

Table A2: Comparative analysis of F-Scores for SRIE training and evaluation corpus