Recognizing Mentions of Adverse Drug Reaction in Social Media Using Knowledge-Infused Recurrent Models

Gabriel Stanovsky[*]
Bar-Ilan University
Ramat Gan, Israel

Daniel Gruhl
IBM Research Almaden
San Jose, California, USA

Pablo N. Mendes[*]
Lattice Data, Inc.
Menlo Park, California, USA

Abstract

Recognizing mentions of Adverse Drug Reactions (ADR) in social media is challenging: ADR mentions are context-dependent and include long, varied and unconventional descriptions as compared to more formal medical symptom terminology. We use the CADEC corpus to train a recurrent neural network (RNN) transducer, integrated with knowledge graph embeddings of DBpedia, and show the resulting model to be highly accurate (93.4 F1). Furthermore, even when lacking high quality expert annotations, we show that by employing an active learning technique and using purpose built annotation tools, we can train the RNN to perform well (83.9 F1).

1 Introduction

Identifying medical concepts in social media narratives is the task of recognizing certain phrases in the context of a user's post. Each phrase is also assigned a label from a set of predefined medical types. For instance, given the sentence "*Aspirin cured my terrible headache, but made me sleepy*", the following medical concepts can be identified: "*Aspirin*" is identified as **Drug**, "*terrible headache*" as a **Symptom** and "*made me sleepy*" should be spotted as a **Adverse Drug Reaction**.

Having an automatic identification process can help domain experts examine large quantities of unstructured data, and quickly identify emerging trends. For example, associating previously unknown side effects with a given drug, or identifying an unforeseen impact to a change in the manufacturing process.

[*]Work performed while at IBM Research Almaden.

There are several challenges in addressing this task. First, *context* is crucial to type assignment. Compare the previous example with "*Aspirin cured my sleepiness but gave me a terrible headache*", while the medical concepts are similar, their context determines their particularly associated type label.

The social media domain poses additional challenges. User narratives on social platforms tend to be non-grammatical, use colloquialisms, slang, and generally informal language. For example, a user may express sleepiness as "*hard time getting some Z's*". This hinders the use of pre-trained statistical parsers or simple string matching techniques.

In this work we focus on the identification of *Adverse Drug Reactions* (ADR). These are unwanted side effects which the user clearly identifies as caused by the intake of a drug. ADRs are particularly challenging to spot, as they can be articulated in a variety of ways and can often be confounded with the symptoms addressed by the drug.

Previous work in this field has mainly used carefully built lexicons and hand-coded rule based systems (Iqbal et al., 2015). While each individual system achieves good results in the particular domain, porting these rules to another domain is non-trivial. For example, identifying psychiatric drug adverse reactions will probably consist of a much different lexicon than that of cardiac medication.

In this work we address ADR mention recognition by with recurrent neural network (RNN) transducers (Graves, 2012). We propose a framework which makes novel use of general, non-task-specific medical knowledge from DBpedia (Lehmann et al., 2015).

Our contributions are two fold: first, we use the high-quality annotation of the CSIRO Adverse Drug Event Corpus (CADEC) (Karimi

et al., 2015) to train accurate models, achieving performance of 93.4 F1 on the CADEC test section. Analyzing the performance of our models, we show that the theoretically unbounded memory of the RNN is good at capturing the context of the narratives, and that external DBpedia knowledge provides additional improvements.

Second, in most medical domains there is no large preexisting collection of expert annotation (i.e., gold standard test and training data) and obtaining one is an expensive prospect. We therefore address the following research question: "How quickly can our system converge on 'good enough' annotations?".

For that end, we use only the test portion of CADEC to test a model trained on non-expert annotation. To expedite this process, we use a purpose built annotation tool in conjunction with an active learning technique to sample the most informative examples to annotate. This approach achieves reasonable results in a very short time (83.9 F1 in only one hour of human annotation). We suggest that this framework is a promising avenue for researches exploring low-resource domains, alleviating the need to first commit to an expensive annotation endeavor.

2 Background

In this section we describe the CADEC corpus, which we use to train and test our model, and DBpedia, along with the recent paradigm of knowledge graph embeddings integrated into our RNN.

2.1 CSIRO Adverse Drug Event Corpus

The recently created CSIRO Adverse Drug Event Corpus (CADEC) (Karimi et al., 2015) contains medical concepts annotation in posts from `Ask a Patient`[1], an online forum collecting medical patient narratives.

For example, a forum entry regarding a certain drug starts with "*I experienced one night of agonising upper stomach pain, diarrhoea and sleeplessness*".

CADEC used `brat` (Stenetorp et al., 2012) to annotate five types of medical concepts: (1) *Drug*, names of medicine or drug, e.g., "*Diclofenac*" or "*Aspirin*"; (2) *Adverse Drug Reaction*, an unwanted reaction which according to the text is clearly associated with taking the drug, e.g, "*acute stomach pain*"; (3) *Disease*, the reason for taking the drug, e.g., "*insomnia*" or "*aggression*"; (4) *Symptom*, manifestations of the disease, e.g., "*trouble sleeping*" or "*constantly angry*"; and finally (5) *Finding*, a clinical finding that does not pertain to any of the above categories.

Each annotation consists of a word span (possibly non-contiguous) and a mapping of the marked span to medical ontologies (SNOMED (Cote et al., 1977), AMT[2], and MedDRA (Brown et al., 1999)).

Each post is annotated by either a medical student or a computer scientist, screened by the authors of the papers, and finally reviewed by a clinical terminologist. The annotations spanned 1,244 posts relating to 12 drugs divided into two groups (medications with Diclofenac as an active ingredient, and Lipitor). Corpus statistics are presented in Table 1.

In the course of this work we will use the CADEC Adverse Drug Reaction annotations to train and test our models.

2.2 DBpedia and Knowledge Graph Embeddings

DBpedia is a large-scale cross-domain multilingual knowledge base extracted from Wikipedia (Lehmann et al., 2015). DBpedia uses a schema with over 320 entity types and 1,600 property types to describe nearly 4 million entities. Besides the common Person, Location and Organization entity types, it also includes descriptions of drugs, diseases, symptoms and disorders, among others.

Using a knowledge graph such as DBpedia

[1] `http://www.askapatient.com`

[2] `https://goo.gl/xRCGPN`

	Train	Test	All
# Posts	935	309	1244
# Sentences	5723	1874	7597
# Words	95979	31855	127834
# Unique Words	5788	3373	9161

Table 1: Statistics for the CADEC corpus (See section 2).

requires an intimate knowledge of its entity and relation types, as well as its subtle representation decisions. This creates challenges with using knowledge graphs in a machine learning (ML) setting, where the signals are coming from different sources and are often normalized and assimilated to make the final prediction.

In order to allow ML algorithms to make use of the information encapsulated in such graphs, a recent line of work (Bordes et al., 2011; Bordes et al., 2013; Wang et al., 2014; Ji et al., 2016) has compellingly suggested to embed entities as d dimensional dense real vectors, and relations as two projection matrices: R^{lhs} and R^{rhs}. Similarly to word embedding techniques (Collobert et al., 2011; Mikolov et al., 2013), a deep learning model is trained to differentiate between observed (positive) and non-observed (negative) triples, by minimizing the following score for positive triples (E_i, R, E_j):

$$||R^{lhs}E_i - R^{rhs}E_j|| \quad (1)$$

During training, the model learns the entity and relation embeddings, desirably encoding some of the semantics and co-occurrence information of the original knowledge graph.

3 Recognizing Mentions of Adverse Drug Reaction

In this section we formally define the task of in-context recognition of Adverse Drug Reactions (ADR) mentions and describe our proposed problem modeling.

3.1 Task Formulation

We follow CADEC's definition for ADR, as described in Section 2.

Formally, we define the ADR mention recognition task as a sentence level chunking task, where each word can either be: (1) Beginning of an ADR span (B); (2) Inside an ADR span (I); or (3) Outside of the span of an ADR (O);

For example (tags in subscript):

"I_O *stopped*$_O$ *taking*$_O$ *Ambien*$_O$ *after*$_O$ *three*$_O$ *weeks*$_O$ – *it*$_O$ *gave*$_O$ *me*$_O$ *a*$_O$ ***terrible***$_B$ ***headache***$_I$"

This formulation, termed BIO tagging (Ramshaw and Marcus, 1995; Sang and Veenstra, 1999), is equivalent to noun phrase (NP) chunking annotation convention with a single type of NP.

While Ratinov and Roth (2009) have shown that a more elaborate tagging scheme (BILOU)[3] improved performance in their experiments in Named Entity Recognition, those experiments are out of the scope for this work.

This task depends heavily upon context, as the same word span can appear as an ADR in one text, and as a Symptom in another. For example, the first entry in Table 2 mentions several ADRs ("*made me gain 30 lbs*", "*made my BP go up so high*", "*gave me more anxiety*") associating each with a different drug ("*Klonopin*", "*Lexapro*", etc.), while the second entry in the table uses some of the same surface forms to refer to an addressed Symptoms (e.g., "*It helped both my anxiety and IBS*").

Furthermore, our model will need to cope with texts from social media which tend to be colloquial, non-grammatical, variably spelled and overall employ highly informal phrasing. The rest of the entries in Table 2 present several snippets from the `Ask a Patient` corpus, illustrating some of these challenges.

3.2 Recurrent Neural Networks Transducer

Formulated this way, and given that the sentences in `Ask a Patient` are of arbitrary length, it seems applicable to model the task using Recurrent Neural Networks (RNNs). This approach was proven to be effective in many recent NLP papers. For a recent and extensive survey of RNNs in NLP see (Goldberg, 2015).

Specifically, we use a bi-directional LSTM transducer (Graves, 2012) which outputs a probability distribution over the three possible labels (B, I, and O) per word, taking into account arbitrary length contexts from both past as well as future words.

Pretrained word embeddings It is common in recent neural networks frameworks to initialize the model's word embeddings with pretrained parameters, from a much larger (often unsupervised) corpus. We experiment

[3]BILOU uses tags for: Beginning, Inside, Last and Unit length chunks.

Text Snippet	Challenges
"*I've tried Klonopin which gave me nightmarish side effects, Lexapro which made me gain 30 lbs and that gave me more anxiety and borderline depression, Effexor which made my BP go up so high I was hospitalized for 4 days (high bp runs in the family), and Buspar which did n't even touch my anxiety.*"	Long post describing Adverse Drug Reactions caused by various previous drugs.
"*It helped both my anxiety and IBS immensely.*"	Describes cure from symptoms (not ADRs).
"*After the second pill the same progression of symptoms only now the abdominal gas, cramps and pain would be with me all day.*"	Coordination leads to non-contiguous ADR spans ("*abdominal cramps*", "*abdominal pain*").
"*I had the usual problems as most people. Driving, buying things online, cooking, eating, sexual activity.*"	The qualifier "while asleep" is implied by "*as most people*" but never explicitly stated.
"*Short term more loss*"	Ungrammatical.
"*started having tension headaches. did not relate to Ambien.*"	The first sentence implies ADRs, while the second negates them.

Table 2: Examples from the `Ask a Patient` forum.

with initializing our word embeddings from both out of domain (and out of the box) word embeddings from Google (Mikolov et al., 2013), as well as with purpose trained embeddings utilizing predicate-argument structure from Open-IE (Etzioni et al., 2008) (following (Stanovsky et al., 2015)) from the Blekko medical corpus (a 2GB corpus of web pages categorized as "medical domain" by the Blekko search engine[4]).

4 Augmenting RNNs with DBpedia

Despite the original motivation for knowledge graph embedding, few efforts were made to use such embeddings as components in larger NLP frameworks. Instead, previous research has focused on embedding techniques, as outlined in section 2. In this section we describe a novel framework which utilizes DBpedia concepts embeddings, in addition to the common use of pretrained word embeddings. We specifically use DBpedia due to its good coverage of our domain of interest. Furthermore, since it relies on Wikipedia, it might also be applicable for non-medical domains. The presented approach, however, is not limited to any particular knowledge base and in future work we plan to extend it beyond DBpedia.

The motivation for using external knowledge bases in our case stems from the relatively small size of the CADEC corpus (see Table 1), in comparison with other neural models training corpora. For example, The Penn Treebank (Marcus et al., 1993) which is often used for training dependency parsing algorithms, consists of roughly 7M tokens, versus only about 95K tokens in CADEC.

4.1 Overriding Word Embeddings with DBpedia Concepts

We augment our model with a pretrained knowledge graph embedding of the "Drug" and "Disease" categories from DBpedia, training as discussed in Section 2.2. When a word in a CADEC entry is a lexical match with one of the DBpedia entities we override its features with the DBpedia embeddings. Intuitively, this framework introduces complex semantic relations between prominent and task relevant words in the `Ask a Patient` posts. For example, DBpedia draws "*Aspirin*" and "*Ibuprofen*" closer in the embedding space as both appear under the "Nonsteroidal anti-inflammatory drug" category (a relation which is modeled in DBpedia). While this changes the embedding of a small subset of the words, these are meaningful and frequently occurring in our setting (see details in Section 6).

Figure 1 shows the complete architecture of our model, including the RNN transducer LSTM and the pretrained word embeddings augmented with DBpedia entity embeddings. The loss from the network propagates back to the word embeddings, allowing them to assimilate task-specific information during training.

5 Human in the Loop

From our experience, real world applications often do not have a pre-existing rich gold standard corpus from which they can efficiently train high quality models. This lack creates a serious impediment to entering and explor-

[4] `https://en.wikipedia.org/wiki/Blekko`

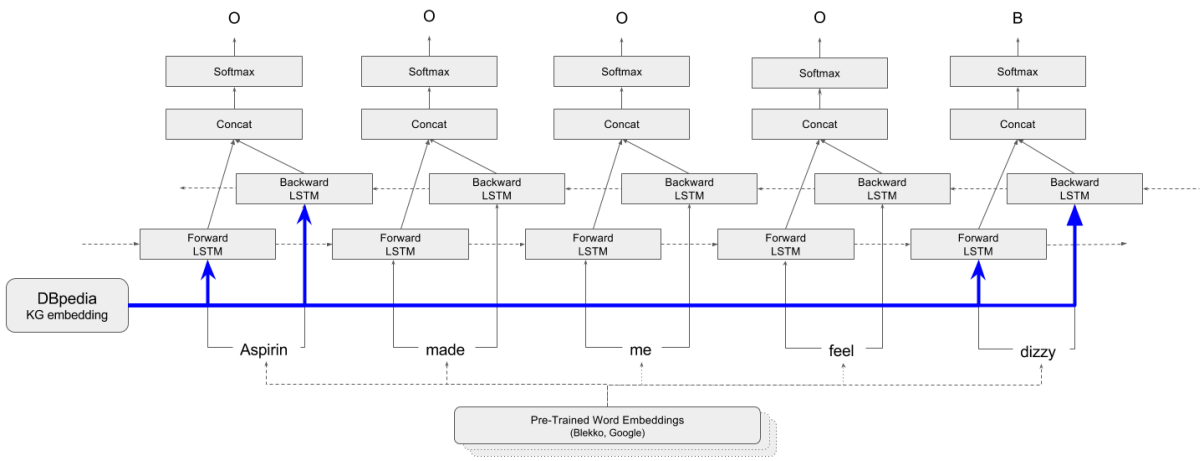

Figure 1: The bi-LSTM transducer, integrated with DBpedia knowledge graph embedding (left). We experiment with several corpora for training external word embeddings (bottom) and override them for DBpedia concepts (e.g., "*Aspirin*", "*dizzy*"). See Sections 3 and 4 for details.

ing the opportunities for text analytics in such domains, due to the high cost of producing the requisite semantic assets.

The medical domain is one where this is especially true. Even within a particular specialty, e.g., oncology, very different information may need to be extracted depending on the type of cancer being explored. In domains where the patient's verbatim comments are critical (e.g., psychiatric, physical rehabilitation) there can be even more variability and ambiguity. For instance, a comment of feeling "*pins and needles*" could be a result of peripheral neurological issues, or a panic attack, and may be expressed with multiple misspellings, punctuation and grammar variations.

To that end, we test our suggested model in a human-in-the-loop approach to gauge how quickly an analytic developer might obtain "good enough" training and test data to develop first generation code and begin to explore the results.

To simulate this for an experiment, we ignore the CADEC training annotations and instead interleave adjudication of small batches (100 sentences) with iterations of model training (see Figure 2). To allow for these fast iterations we need to provide solutions in several areas. For quicker annotation, we developed RASCAL (Rapid Adjudication of Semantic Classes to Accelerate Learning), a purpose built annotation tool which expedites the annotation and adjudication process. Furthermore we employ an active learning technique to focus the human adjudicator's time on examples that the model finds most confusing.

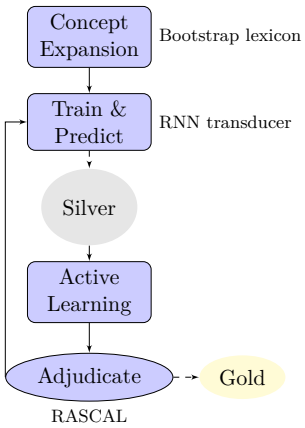

Figure 2: Train-Predict-Adjudicate loop

This process, as elaborated below, has two outcomes: (1) Expedited machine-assited production of an annotated gold standard; and (2) Rapid training of high precision models due to the active learning technique.

5.1 Bootstrap

While the learning process can be initiated by simply beginning to annotate the corpus, we find a more rapid start up is achieved by employing an extensive lexicon of Adverse Drug Reaction phrases. Fortunately, lexicon expansion techniques (Coden et al., 2012) provide a way to rapidly bootstrap this portion of the problem. Typically the user provides a few ($\sim$ 3) examples of the type desired, and the system comes back with dozens of suggestions of potential new phrases. The user approves or rejects these and the system repeats with this additional knowledge. The process typi-

cally generates a couple of hundred candidate terms in a few minutes.

5.2 Active Learning

The bootstrap lexicon can then be used to obtain a preliminary noisy CADEC training set for the RNN, by marking each occurrence of a lexicon term as an ADR. After we train our model, we want to choose informative samples to adjudicate and refine our training set (and subsequently, our model) in the next iteration.

A process in which a model chooses its next training examples is often referred to as *active learning*, and is a well researched area of machine learning (see (Settles, 2010) for an extensive survey). For our purposes, we use the uncertainty sampling criterion (Lewis and Gale, 1994). Intuitively, this ranks the samples according to the model's belief it will mislabel them.

Formally, we sort all samples (x, y) according to the following measure:

$$1 - \Pr_{\theta}(\hat{y}|x) \tag{2}$$

Where:

$$\hat{y} = \underset{y \in \{B,I,O\}}{\operatorname{argmax}} \Pr_{\theta}(y|x) \tag{3}$$

We choose the top 100 samples according to this metric, and adjudicate them, as described below. In order to assess the impact of this step we also perform the same process with random sampling of 100 sentences at each iteration.

5.3 Adjudication

At this stage, a human adjudicator examines the sentences chosen in the previous phase. They then either accept or reject each automatically recognized ADR span in these sentences. Additionally the annotator can mark new spans that were missed.

While the `brat` annotator (Stenetorp et al., 2012) is a popular tool for creating and modifying annotations, it is a bit cumbersome and error prone for tasks such as the one outlined in CADEC.

Notably, the CADEC creators mention that the `brat` annotations required an additional cleaning phase. For example `brat` annotates character spans instead of aligning marked spans to the word level (e.g, an annotator might wrongfully mark "*pai*" instead of "*pain*"). While some of `brat`'s more complex features are a good fit for other annotation tasks, they reduce agility and do not add much value in ours. We therefore implemented a simple rapid adjudication system (code-named RASCAL) that is tuned to the particular task of adjudicating and adding annotations in the context of ADR mention recognition.

RASCAL introduces simplifications such as single click removal of incorrect annotations, automatic alignment of the spans to include whole tokens, and single key "approve and move to next document" support. This results in very fast annotation times. As the system improves its understanding of the entity to tag, much of the annotator's time is spent simply approving annotations, with about one in four requiring their addition of a missed span.

Over the 1,100 annotations we found an average time of about 3 seconds per sentence. In a controlled experiment comparing the annotation of 100 sentences in `brat` versus RASCAL, this represented at least a four fold improvement over comparable `brat` times without pre-annotation. Since we find `brat` is slower at removing errors and entering split annotations we anticipate the discrepancy may be even higher with pre-annotation.

This improvement does have a cost, however; RASCAL only allows a single annotation type at a time, so the annotation of two predefined types (e.g., Drug and ADR) requires two passes. Second, RASCAL does not support non-contiguous span annotations. This is of especial trouble when there are coordinated spans (e.g., "*my neck and back are both spasming*" should be "*neck spasming*" and "*back spasming*").

While these were uncommon in our corpus (see detailed analysis in Section 6), it does suggest that perhaps performing some of the annotations with `brat` after doing the initial ones with RASCAL might help improve precision if desired.

5.4 Repeat

Given these adjudicated annotations, we can refine our bootstrap lexicon (with the newly acquired ADR mentions) and automatically re-annotate the entire training corpus accord-

ing to it, generating a new iteration of the training data which closes the loop back to training and predicting (Section 5.2).

Knowing when to stop is always a challenge with learning systems. For the sake of these experiments we chose to stop after an hour of an annotator time (the initial lexicon expansion bootstrap and annotating/adjudicating 1,100 sentences).

However the human annotator using RASCAL gets a fairly good sense of what kind of annotations are being spotted and what is being missed. By looking at the net change in the pre-annotation to post-annotation spans for an iteration it is possible to get a sense of when the learning is leveling off.

In future work, it may also be possible to look at the total uncertainty the RNN finds in the training corpus before and after a training session as a measure of how much more productive learning there may be left.

6 Evaluation

In this section we evaluate our recurrent neural transducer in two scenarios: (1) Using the high quality annotations of the CADEC corpus and (2) Simulating a task with low resources in an active learning scenario and using RASCAL for non-expert annotations, as described in Section 5. Results are shown in Table 3.

Label imbalance The CADEC corpus is imbalanced between the different labels, assigning the label "O" (Outside) to 87.34% of the words. This is due to the fact that most of the text in the `Ask a Patient` forum describes background situation and is not directly related to an ADR (see, for example, the first entry in Table 2). This poses a problem for training accuracy oriented models, as such imbalanced class distribution discourages the learning process to move from a model which assigns a constantly higher probability to the Outside label, regardless of the input sentence (He and Garcia, 2009). Subsequently, this leads to trivial solutions which achieve 0% ADR recall (as no ADRs are retrieved) and 100% ADR precision (as there are also no false positives).

To address this problem we use the SMOTE (Synthetic Minority Over-Sampling) technique (Chawla et al., 2002) which skews the sample distribution by oversampling the minority classes (B and I) during training to get a synthetically balanced training set.

6.1 Experimental setup

We implemented the bi-LSTM transducer model using the Keras framework (Chollet, 2015) with a TensorFlow backend (Abadi et al., 2015). Open IE word embeddings (300 dimensions) were trained on Blekko medical corpus (1 billion tokens) using Open IE 4[5] and Word2Vec (Mikolov et al., 2013), as described in (Stanovsky et al., 2015). For DBpedia embeddings (300 dimensions), we used the code published in (Nickel et al., 2015). We used the code published in (Lemaître et al., 2016) for SMOTE class resampling. Finally, we used the `libact` library (Yang et al., 2015) for the active learning sampling. All models were trained for 100 epochs.

6.2 Results

Several observations can be made based on the results of our experiments (Table 3):

RASCAL achieves good results at a fraction of the annotation effort - RASCAL results are obtained after just 11 cycles of annotation by single annotator (roughly an hour of work), and are then tested against the independently annotated test set of CADEC. The performance of RASCAL is a promising indication that adequately performing models can be obtained very quickly using our framework, when moving to a new annotation task where training data is scarce.

External knowledge improves performance in both scenarios - As can be seen from the ablation test in Table 3, in both supervised and annotator development settings, our pretrained embeddings improve performance by at least 13 points in F1, with a significant edge to Blekko embeddings. This is in part due to its better coverage of the CADEC lexicon, only 408 (7.05% of the CADEC lexicon) unique words were Out Of Vocabulary (OOV) using Blekko, compared with 724 (12.51%) OOV words using Google's embedding. DBpedia provides embeddings for 232

[5] `https://github.com/allenai/openie-standalone`

words (4%) and further adds 2-4 points in both recall and precision.

Uncertainty sampling boosts the learning rate - Figure 3 shows the progression of the best performance obtained at each training iteration. The uncertainty sampling (see Section 5.2) boosts the learning curve, achieving models performing around 80 F1 after just 25 minutes of RASCAL annotation (note the red vertical dotted line).

The relatively small change when increasing the number of annotated instances from 400 to 800 (i.e., before and after the vertical dotted line) is probably due to the long tail nature of the problem: active learning chooses the most prominent examples first, then there is a sharp decline in the novelty of the chosen examples. Further experimentation with active learning techniques may improve performance, yet this falls out of the scope of this paper, and is left as a topic for future research. Overall, it can be seen from Figure 3 that our active learning technique is indeed already superior to random sampling (notice the brown dotted line indicating performance with random sampling).

Context matters - We tested an oracle ADR baseline which had access to the lexicon of all of the ADRs in CADEC. This oracle ignored context and marked every occurrence of a phrase from the lexicon as an ADR. As can be seen in Table 3 (ADR Oracle), this baseline obviously achieves 100% recall, yet, more interestingly, it achieves only 55.2% in precision. Thus in 44.8% of the cases the surrounding context negated the ADR phrase (for example, see the last entry in Table 2).

6.3 Error analysis

In analyzing the RASCAL model, we find that it relatively lacks in recall. This is due to our limited annotation effort having predictably limited coverage. Examining our annotations, we find 449 unique ADRs annotated in RASCAL out of the total 3685 unique ADR phrases in the full CADEC annotation. The RNN model is in fact able to generalize these mentions and find approximately 75% of the mentions, yet it is likely that having a larger RASCAL training set would help improve the coverage of our model. Furthermore, the design choices made in RASCAL trade annotation speed with accuracy. As mentioned in Section 5.3 RASCAL is currently unable to annotate non-contiguous spans, which account for 1005 (15.9%) of the ADRs annotated in CADEC.

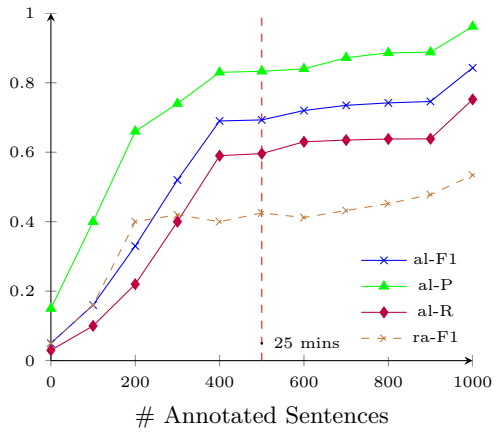

Figure 3: Model performance per annotation cycle (see Section 6). Solid lines represent precision, recall and F1 for active learning, and the dashed line represents F1 for random sampling (precision and recall follow the same trend and are omitted for clarity).

	P	R	F1
String matching	78.6	37.0	50.3
RASCAL	64.0	53.8	58.0
RASCAL+ Google	88.4	68.2	76.8
RASCAL+ Blekko	91.3	70.0	79.4
RASCAL+ DBpedia+ Blekko	**96.2**	**75.2**	**83.9**
ADR Oracle	55.2	100	71.1
CADEC	69.6	74.6	71.9
CADEC + Google	85.3	86.2	85.7
CADEC + Blekko	90.5	90.1	90.3
CADEC + DBpedia+ Blekko	**92.2**	**94.5**	**93.4**

Table 3: Performance of the different baselines by training from RASCAL annotations (top) vs. CADEC training data (bottom). See Section 6 for more details.

Finally, both of our models predict BIO word labels at the sentence level which in some cases does not provide enough context to arrive at the correct label. See, for example, the bottom example in Table 2, in which a very probable ADR phrase in the first sentence ("*tension headaches*") is negated in the second sentence ("*did not relate to Ambien*").

7 Related Work

To the best of our knowledge, there has been no previous work attempting to recognize in-context adverse drug reaction mentions on the

CADEC corpus. There are, however, several papers which addressed the same task on a different corpus, and others who have used the CADEC corpus for orthogonal tasks. In this section we survey two such recent papers.

Limsopatham and Collier (2016) have used CADEC for the normalization of medical concepts. They take as input an out-of-context ADR (e.g., "*I couldn't sleep all night*" or "*head explodes*") and predict its normalized form (e.g., "*insomnia*" or "*headache*", respectively), based on a predefined vocabulary. They use an RNN model and report accuracy of 79.98. This task can be seen a subsequent task to ours. The ADR spans we output can serve as an input for ADR normalization, giving medical experts a consolidated summary of the reported adverse events.

Iqbal et al. (2015) share our motivation to identify ADR mentions in the context of electronic health records (medical correspondence, discharge letters, etc.), which are more formal, as opposed to our focus on social media domain. They take a rule based approach, and come up with an expert built lexicon, which achieves 85 F1 on their test set.

While their approach is carefully built to the specific data set, we show the portability of our model by testing both in a supervised scenario as well as in annotation development scenario.

8 Conclusions and Future Work

We presented a novel model which consists of an LSTM transducer RNN augmented with external knowledge from medically oriented Web crawl and a knowledge graph embedding of medical entities in DBpedia. We showed that the model achieves good results (93.4 F1) when trained and tested on the CADEC corpus.

Furthermore, ignoring the CADEC training data, we showed that through active learning and a task-dedicated annotation tool we can get a reasonably performing model (83.95 F1 on CADEC's test set) with just an hour of annotation effort. This suggests a promising methodology for researchers wanting to explore new domain annotations, without first committing to a heavyweight and expensive annotation effort.

Future work may make further use of the CADEC annotations (e.g., for multi-task learning or concept normalization), and extend RASCAL to get better recall and allow for non-contiguous and multiple label annotations.

Acknowledgments

We would like to thank the IBM Almaden Intelligence Augmentation Team (Alfredo Alba, Linda Kato, Chris Kau, Joe Terdiman and Steve Welch) for many fruitful discussions, and the anonymous reviewers for their helpful comments.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *Conference on Artificial Intelligence*, number EPFL-CONF-192344.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795.

Elliot G Brown, Louise Wood, and Sue Wood. 1999. The medical dictionary for regulatory activities (meddra). *Drug Safety*, 20(2):109–117.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

François Chollet. 2015. keras. `https://github.com/fchollet/keras`.

Anni Coden, Daniel Gruhl, Neal Lewis, Michael Tanenblatt, and Joe Terdiman. 2012. Spot

the drug! an unsupervised pattern matching method to extract drug names from very large clinical corpora. In *Healthcare Informatics, Imaging and Systems Biology (HISB), 2012 IEEE Second International Conference on*, pages 33–39. IEEE.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Roger A Cote, College of American Pathologists, et al. 1977. *Systematized nomenclature of medicine*. College of American Pathologists.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. 2008. Open information extraction from the Web. *Communications of the ACM*, 51(12):68–74.

Yoav Goldberg. 2015. A primer on neural network models for natural language processing. *arXiv preprint arXiv:1510.00726*.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

Haibo He and Edwardo A Garcia. 2009. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284.

Ehtesham Iqbal, Robbie Mallah, Richard George Jackson, Michael Ball, Zina M Ibrahim, Matthew Broadbent, Olubanke Dzahini, Robert Stewart, Caroline Johnston, and Richard JB Dobson. 2015. Identification of adverse drug events from free text electronic patient records and information in a large mental health case register. *PloS one*, 10(8):e0134208.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2016. Knowledge graph completion with adaptive sparse transfer matrix. In *Proceedings of AAAI*.

Sarvnaz Karimi, Alejandro Metke-Jimenez, Madonna Kemp, and Chen Wang. 2015. Cadec: A corpus of adverse drug event annotations. *Journal of biomedical informatics*, 55:73–81.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6(2):167–195.

Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. 2016. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *CoRR*, abs/1609.06570.

David D Lewis and William A Gale. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc.

Nut Limsopatham and Nigel Collier. 2016. Normalising medical concepts in social media texts by learning semantic representation. ACL.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2015. Holographic embeddings of knowledge graphs. *arXiv preprint arXiv:1510.04935*.

Lance A Ramshaw and Mitchell P Marcus. 1995. Text chunking using transformation-based learning. *arXiv preprint cmp-lg/9505040*.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155. Association for Computational Linguistics.

Erik F Sang and Jorn Veenstra. 1999. Representing text chunks. In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, pages 173–179. Association for Computational Linguistics.

Burr Settles. 2010. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11.

Gabriel Stanovsky, Ido Dagan, and Mausam. 2015. Open IE as an intermediate structure for semantic tasks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107. Association for Computational Linguistics.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, pages 1112–1119. Citeseer.

Yao-Yuan Yang, Yu-An Chung, Shao-Chuan Lee, Tung-En Wu, and Hsuan-Tien Lin. 2015. libact: Pool-based active learning in python. Technical report.