Concept Linking for Clinical Text

Justin Fu
justinfu@stanford.edu

Abstract

Concept linking, or linking spans of text to concepts in a knowledge graph (KG), is an important first step in tapping into the wealth of information stored within KGs. Use of terminologies and vocabularies is especially prominent in the biomedical domain, where significant engineering effort has been made to ensure that rich KGs with standardized vocabularies are available.

In this project, we present an approach based on recurrent neural networks (RNNs) that learn mappings from textual spans to concepts of a knowledge graph. Our method achieves generalization by using vector-based representations for concepts, and predicting all concepts along a hierarchy traversal, which is a richer prediction target that can also enable the model to gracefully handle unseen concepts.

Our main result is that we are successfully able to train a RNN-based model to mimic a complex set of morphological and syntactic transformations applied by a state-of-the-art rule-based system, and generalize better than the rule-based system on concepts not present during training time. The RNN also learned embeddings of phrases which are invariant several of the phenomena required to normalize text, such as word order inversion, synonym replacement, and noise in the span boundary. Unfortunately, we are not able to outperform the rule-based method on the original task when the train and test sets overlap.

As part of this project, we also present a method to automatically label a large dataset for concept linking in clinical text, which opens new opportunities to apply machine learning and deep learning methods in this domain.

1 Introduction

Knowledge graphs (KGs) codify a large amount of knowledge in a symbolic representation that is easy for algorithms to use and perform inference on. However, an inherent disadvantage of knowledge graphs is that they are generally hand-engineered by many people, relatively brittle due to their symbolic nature, and utilizing the encoded knowledge is not a trivial task. Several challenges in using knowledge graphs include having concepts that are either too specific or too general, inconsistencies in the same property is represented in different parts of the graph, and concepts with ambiguous names. Moreover, knowledge graphs generally only grow by adding new concepts and relationships, as refactoring is very expensive.

In this project, we tackle the problem of concept linking using recurrent neural networks (RNNs) with a specific focus on being able to generalize to unseen concepts at test time, by using vector-based concept representations that share features and predicting the entire traversal through the hierarchy as our target. The advantage of predicting traversals is two-fold: first, the model fails more gracefully since it can fall back to predicting a coarser concept, and second, the model becomes more interpretable since the model is forced to produce multiple partial predictions.

We apply our method to the medical domain, where a significant amount of data is recorded in plain text medical records written by doctors while treating patients. Concept linking is a common

pre-processing step used for many downstream processing tasks such as search, indexing, and featurization. This in turn is used for tasks such as cohort selection, diagnosis and phenotyping, etc. Both model interpretability and robustness to noise are invaluable in the medical domain. The ability to generalize to unseen concepts is also important because annotated data is very scarce, so it is impossible to see every concept during training, and the knowledge graph does not always have the correct granularity to represent a concept.

2 Background

As described earlier, concept linking is an integral part in biomedical informatics pipelines. A popular and fast method is simply exact string matching against a database of synonyms [1], which has the advantage of very high precision but unfortunately also suffers from low recall. The ShARe/CLEF eHealth Challenge was a recent [4] (2013) competition which benchmarked two tasks: span recognition and concept linking. Here, a rule-based algorithm [8] which applies various transformations such as stemming, suffix replacement, acronym expansion, etc. achieved state-of-the-art results on medical record span recognition. Like exact string matching, rule-based methods achieve high precision but struggle with tasks requiring softer reasoning such as synonym replacement ("bleeding" vs "hemorrhage").

Due to our goal to generalize to unseen concepts during test time, our new problem formulation is similar to paraphrase and semantic similarity tasks, in which people have applied methods such as siamese convolutional networks [5] and tree-structured RNNs [9].

Our models in particular are based on recurrent neural networks (RNNs) and sequence-to-sequence models, which have proven to be effective in tasks such as machine translation [6] and parsing [10].

In our project, we use Gated recurrent units (GRU) [2], which are a specific form of recurrent neural network which contain architectural modifications to mitigate the vanishing gradient problem. The forward propagation equations for GRUs are specifically:

Initial state:

$$h_0 = 0$$

Gates:

$$z_t = \sigma(W^z x_t + U^z h_{t-1})$$
$$r_t = \sigma(W^z x_t + U^z h_{t-1})$$

Output:

$$c_t = \tanh(W x_t + r_t \odot U h_{t-1})$$
$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot c_t$$

Where $x_1, ...x_T$ is the input sequence, and $\odot$ represents element-wise multiplication.

3 Approach

3.1 Data

We have two primary sources of data - the knowledge graph (from SNOMED-CT), and the input-output pairs of our model, which are spans of text and annotated concepts, respectively (from either ShARe/CLEF or Synthetic).

3.1.1 SNOMED-CT (Systematized Nomenclature of Medicine – Clinical Terms)

SNOMED-CT is a general, publicly available medical knowledge graph. We use a 182,719 concept subset of SNOMED which includes diseases, symptoms, and other medical findings. Our main use of SNOMED is it's concept hierarchy ("is-a" relationships). To remove SNOMED's DAG inheritance structure, we deterministically select the node with the most leaf nodes if a node has multiple parents.

An example of SNOMED's hierarchy can be seen in figure 1.

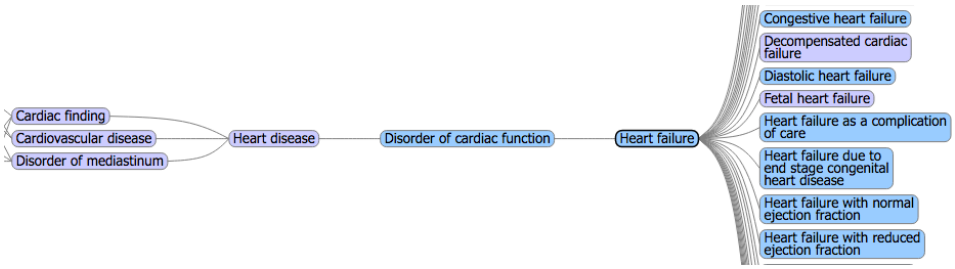

Figure 1: An excerpt of SNOMED's hierarchy for the concept "Heart failure". SNOMED has a DAG inheritance structure, as seen by the 3 parents of "Heart disease".

```
Brief Hospital Course:
53 yo woman with history of [CAD:Coronary Artery Disease] and
[CHF:Congestive Heart Failure].s/p 6 weeks IV antibiotics
presents with [arthritis:Arthritis], [blood in stomach:Gastric Hemorrhage].
```

Figure 2: An typical example of our synthetic data. Labeled spans are highlighted in blue, with the gloss/mention span to the left of the colon and the name of the concept on the right.

3.1.2 MIMIC-III (Medical Information Mart for Intensive Care)

MIMIC-III is a collection of roughly 50,000 medical records collected by doctors from an ICU, provided by the MIT Lab for Computational Physiology. These notes are raw text, and lack annotations and concept labels. We use this text as part of our synthetic dataset (see section 3.1.5).

3.1.3 ShARe/CLEF eHealth Challenge 2013

The ShARe/CLEF dataset is 297-note a subset of MIMIC which contains spans and human-labeled annotations. Since this is our only source of gold labels, we use this dataset in our evaluations. ShARe/CLEF also contains spans marked as "CUI-less", meaning they do not correspond to an exact concept in SNOMED, but from our observations, there is often a reasonable coarser concept to assign. There are roughly 11167 annotated spans in total, 3374 which as "CUI-less".

ShARe/CLEF is a subset of a SemEval 2015 challenge, but out application to use that dataset was unsuccessful.

3.1.4 UMLS Metathesaurus (Unified Medical Language System)

UMLS is a conglomeration of medical lexicons and ontologies (including SNOMED-CT). The metathesaurus provides correspondences between concepts in the various sources, which gives us a set of synonyms for each concept. These synonyms are used in the rule-based system we use to generate additional training data (see section 3.1.5).

3.1.5 Synthetic Dataset

In order to generate additional data, we applied a rule-based method based on [8] to MIMIC-III in order to generate roughly 80,000 unique spans with concept labels. We run a sliding window (of sizes 1, 2, 3, and 4) over raw text, and create a (span, label) data point if the rule-based method identifies a concept. When compared with gold spans in ShARe/CLEF, we get 89.4% precision and 56.5% recall, excluding CUI-less concepts, making this a reasonable method. We would rather to err on the side of high precision and low recall than high recall and low precision.

Additionally, we apply the expansion rules to the synonym sets from the UMLS Metathesaurus - these include inserting stopwords, suffix transformations (ex. "dilated" to "dilation"), word order inversion, etc. to generate another set of training examples. This is similar to data augmentation done in other fields such as vision (ex. rotating and translating images).

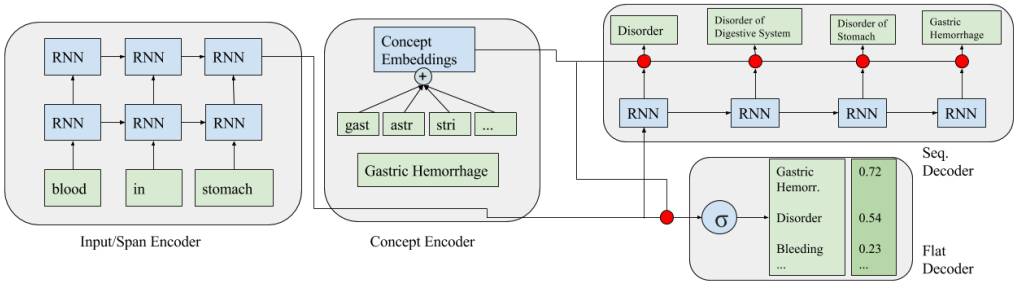

Figure 3: A high level diagram with all components of our model. The red dots represent inner products. Note that only one of the decoders is used at a time.

3.2 Model

We divide our model into 3 main components: the span encoder, the concept encoder, and the decoder. The inputs to our model are the spans of text, and the labels are a list of concepts in the hierarchy representing a traversal (we start at the root concept and end on the actual concept).

A high level diagram of our model shown in figure 3.

3.3 Span Encoder

Our span encoder is a 2-layer, 256 unit LSTM which reads as input the input span represented as word vectors (pretrained with GloVe [7]), and produces a 256-dimensional span embedding e_s as its output, which corresponds to the hidden state of the last timestep.

Let x_t denote the sequence of word vector inputs.

$$h_{1,t} = \textbf{GRU}_1(x_t, h_{1,t-1})$$
$$h_{2,t} = \textbf{GRU}_2(h_{1,t}, h_{2,t-1})$$
$$e_s = [h_{1,T}, h_{2,T}]^T$$

3.4 Concept Encoder

The concept encoder produces a 256-dimensional embedding e_c for each concept by averaging the embeddings of its 4-grams (the 4-gram embeddings are randomly initialized). We then stack these vectors as a matrix E_c.

Let the ngrams of a concept be labeled n_1 through n_k:

$$e_{ci} = \textbf{mean}(n_{i1}, n_{i2}, ...n_{ik})$$
$$E_c = [e_{c1}, e_{c2}, ...]^T$$

This was originally a copy of the span encoder to form a Siamese network, but training was too slow, even on toy data, due to the large concept vocabulary. We also attempted to incorporate the other major source of information that could provide generalization, the structure of the hierarchy, by incorporating a "routing" vector which was a function of the concept vectors of a concept's children, but this also turned out to be infeasible computationally.

3.5 Decoder

We implemented two possible decoders, which read the span and concept embeddings to predict output concepts. For both decoders, we use a sampled cross-entropy loss function (flat decoder uses 0-1 cross entropy, sequence decoder uses multi-class cross entropy) summed over training examples, and averaged across timesteps in the case of the sequence decoder.

$$L = \sum_{i=1}^{N} CE_{sampled}(y_i, \hat{y}_i)$$

3.5.1 Flat Decoder

Our first decoder simply takes the dot product of each span (after projecting it to 256 dimensions) and concept embedding, and feeds each into a sigmoid to produce a probability for each concept.

$$e_{s,proj} = W_{proj} e_s$$

$$\hat{y} = \sigma(E_c e_{s,proj})$$

In this decoder, the label is a sparse vector with ones in locations corresponding to concepts along the traversal. Thus, this model has no notion of the order in the traversal. In order to recover a traversal, we can apply a greedy decoding method [3] which repeatedly accepts the highest scoring concept if it is consistent with the previously selected concepts, and then filling in the blanks to complete the traversal if necessary.

3.5.2 Sequence-to-Sequence Decoder

The second decoder involves running a second 2-layer, 256-unit LSTM during decoding to produce a sequence of concepts corresponding to the traversal. The hidden state of the each layer is initialized with the final hidden state of the the respective layer in the span encoder.

$$h_0 = e_s$$

$$h_{1,t} = \mathbf{GRU}_1(x_t, h_{1,t-1})$$

$$h_{2,t} = \mathbf{GRU}_2(h_{1,t}, h_{2,t-1})$$

$$\hat{y}_t = m \odot h_{2,t}$$

Where m is a mask which contains 1 in locations corresponding valid children along a traversal (only used during test time).

3.6 Data phenomena

Here we describe in detail some of our intuitions about the phenomena in the dataset, and how our approach addresses them:

1. Synonyms (Ex. "lung" vs "pulmonary") - We use pretrained word vectors as part of our input representation.
2. Morphology (Ex. "left ventricular dilation" vs "left ventricle dilated") - We use n-grams as part of our input representation.
3. Acronyms and abbreviations (Ex. "CHF" vs "Congestive Heart Failure") - We rely on the synthetic dataset to enumerate these in the training set.
4. Word order/stop words (Ex. "pain in chest" vs. "chest pain"). We also rely on the synthetic dataset to enumerate these out, and hope that the model to generalizes.
5. Context (Ex. "PLT" vs. "platelet" vs "primary lymphocyte test") - We ignore this case since it is actually rare in our dataset. If we were to tackle this problem, we can simply create embeddings for the context in a similar manner to the mention span.

3.7 Implementation

Our model was implemented in Tensorflow, and trained on a computer with an NVIDIA GTX 970 graphics card with 4GB of RAM. Training each model took approximately 8 to 12 hours.

Model	Synthetic	ShARe/CLEF	S/C New Concept	S/C New Concept (Relax)
Flat Decoder	96/94/95	80/72/76	40/25/31	60/33/43
Seq2Seq Decoder	93/94/93	40/7/13	10/2/3	22/10/14
Sieve Model [8]	100/100/100	99/91/95	98/8/15	98/8/15

Table 1: Precision/Recall/F1 scores for each model on each evaluation task.

4 Experiments

We perform evaluation for 3 models (Flat decoder, Sequence decoder, and the rule-based sieve model [8] baseline) on 4 different tasks:

1. Synthetic: This evaluation measures performance on our synthetic dataset generated by the rule-based sieve model. We train on 700,000 spans and test on 300,000 spans.
2. ShARe/CLEF: This evaluation measures performance on the original 2013 challenge. There are 199 train notes, and 99 test notes. The train set contains 5816 spans, and the test contains 5351.
3. ShARe/CLEF New Concepts: This evaluation measures performance when the test concepts are excluded from the train set. All models have access to the synthetic dataset in this case.
4. ShARe/CLEF New Concepts, Relaxed: This evaluation measures new concept performance, except we allow the models to predict up to 5 concepts, and count any as correct if they are within 2 steps according to SNOMED's concept hierarchy.

Results for each evaluation are shown in table 1.

Unfortunately, while the sieve model is publicly available on GitHub, we were unable to obtain code for any other systems for the ShARe/CLEF challenge, and thus lack other baselines to compare against.

4.1 Analysis

As expected, both of our neural network models perform poorly on the original task, likely because of extreme lack of data (only 5800 training examples, compared to the equal sized test set). Thus, it would be expected that the rule-based method performs very well.

The sequence decoder performed poorly, most likely because its task is much harder (it must predict the traversal in order), and more importantly, the concept embeddings currently lack information pertaining to the hierarchy. As a toy example, consider seeing the span "dog" and trying to predict if it is a "mammal" or a "reptile". If the "mammal" concept vector looks like the "dog" concept vector, this task would be easy, but with our embedding method we cannot look ahead to the children of "mammal". Thus, it is forced to memorize these relationships without opportunity for generalization.

The flat decoder model showed promising ability to mimic the rule-based system, as evidenced by its high accuracy on the synthetic evaluation. A PCA plot showing our span embeddings is shown in figure 4. It also shows promising ability to generalize in the new concept setting, but the accuracy is still low and would not be usable in a real-world setting.

5 Conclusions and Future Work

In this project, we have presented a recurrent neural-network based method for concept linking on clinical text. We also automatically construct a large, supervised dataset for this task when previously only several thousands of annotated examples were available, which is woefully inadequate for machine learning systems when there are over 100,000 possible labels. While we have not yet achieved performance that is usable on a real world task, the model seems promising and there are many places where we cut corners that could result in greatly increased performance.

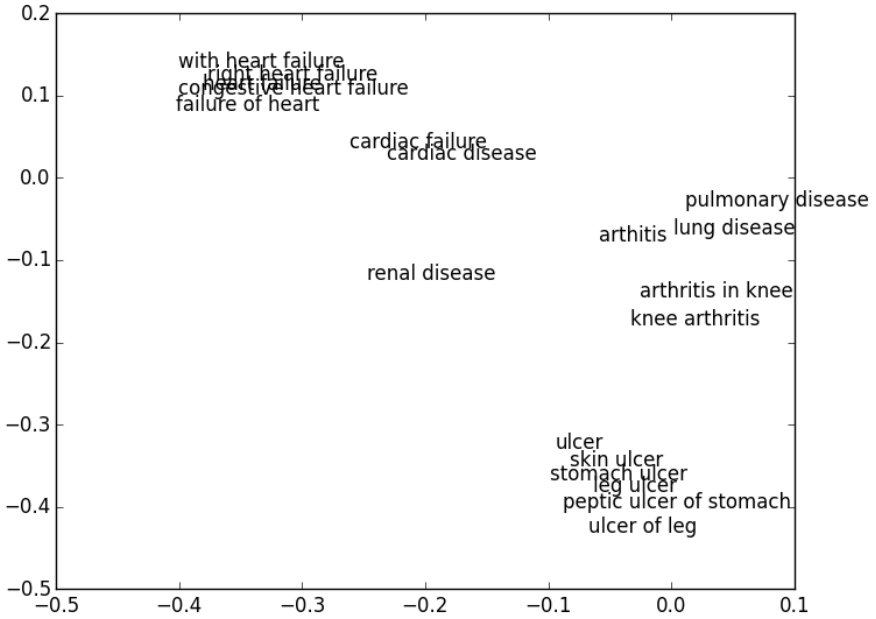

Figure 4: 2-dimensional PCA projections of span embeddings generated by the flat decoder model. The encoder shows moderate ability generalize to word order, stopwords, synonyms, etc.

The most significant component that could be improved is the concept embedding. We currently use a very crude 4-gram average which ignores ordering and context around the n-grams, and information from the hierarchy. As mentioned in the modeling section, we tried a Siamese network approach to the concept embeddings and a "routing" vector approach where each parent's concept vector was a function of its children, but these incurred significant computational cost since the number of concepts is so large. A possible avenue for further investigation is on how to make efficient approximations to these procedures, which could provide richer vector-based representations for concepts.

Another interesting aspect we ignored was the DAG-structure of the hierarchy. For example, the stomach is both a structure in the digestive system and a structure in the abdomen. This means there are multiple correct paths to leaves, which breaks the greedy decoding method of [3]. A DAG structure could help decoding methods similar to beam search, since multiple paths converging to one concept would boost confidence for that concept.

A third path we can explore is a bootstrapping approach to learning our model in an unsupervised fashion. We already perform the initial labeling using a rule-based method, but it may be possible to increase the quality of the training data using a principled bootstrapping method.

References

[1] C. Youn C. Callendar M. Storey M. Musen C. Jonquet, N. Shah. Ncbo annotator: semantic annotation of biomedical data. *International Semantic Web Conference*, 2009.

[2] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

[3] D. Gillick D. Yogatama and N. Lazic. Embedding methods for fine grained entity type classification. *ACL-IJCNLP*, 2015.

[4] H. Suominen, et al. Overview of the share/clef ehealth evaluation lab 2013. *Information Access Evaluation. Multilinguality, Multimodality, and Visualization*, 2015.

[5] H. He, K. Gimpel, and J. Lin. Multi-perspective sentence similarity modeling with convolutional neural networks. *Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

[6] Q. Le I. Sutskever, O. Vinyals. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[7] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[8] J. Souza and V. Ng. Sieve-based entity linking for the biomedical domain. *Proceedings of ACL-IJCNLP Volume 2: Short Papers*, 2015.

[9] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. *CoRR*, abs/1503.00075, 2015.

[10] O. Vinyals, L. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. E. Hinton. Grammar as a foreign language. *CoRR*, abs/1412.7449, 2014.