Sequence tagging for medical problems detection

Yacine Jernite

January 17, 2016

1 Introduction / Reminders

Our ultimate goal is to identify mentions corresponding to medical problems in text and map them to UMLS CUIs. Our initial two-steps approach was to use a simple CRF for mention detection (step 1), and focus on a more elaborate method to map mentions to CUIs, or even identify mistakes of the detection (step 2).

The reasoning behind this approach was that we felt we could make use of more interesting techniques for the second step, including leveraging MIMIC text for semi-supervised learning, and using the MRF model we developed for our ICML paper as a prior on the CUI distribution. To that end, I built a model predicting a CUI given a mention, its neighbouring words and neighbouring mentions. I first tried to do semi-supervised learning without using the MRF prior by training on MIMIC matches with different kinds of noise. I was unable to get any improvement from the MIMIC data.

2 Data Description / State of the Art

We are working on the Semeval task for analysis of clinical test. The two most recent editions of the challenge are 2014 [6] and 2015 [2].

The 2014 edition had a training + development set of 298 notes (199 + 99) and a test set of 133 notes. In 2015, these were merged into the training + development set, and 100 notes were added as a test set (we only have access to the 298 + 133, Noemie Elhadad told us the team still had plans for a challenge involving the test set).

In 2014, the challenge was divided into task 1a (identification of disorder mention span), and task 1b (mapping to UMLS, our step 2). Task 1a was evaluated for F measure, and task 1b for accuracy (recall: number of mentions correctly identified and mapped divided by number of gold mentions). In 2015, both tasks were merged, and only the global F measure was reported.

The best result for task 1a in 2014 came from the UTH_CCB team [10]. They achieved an F measure of 81.3 using an ensemble of CRF and Structured SVM, using UMLS, POS, cTakes output, Brown clustering and random indexing features and a {B, I, O, DB, DI, HB, HI} tag set. Their F measure on the global task was 71.8 (re-calculated fro their reported "accuracy" measure).

In 2015, we only have the F measure on the global task for the fifth best team, TAKELAB [3]. Their best development set F measure is 73.0 (with a test set F measure of 72.7). Their system is a CRF with BIO tags, post-processing for disjoint and overlapping mentions, and a set of features including SNOMED similarity and informative ngrams. The best F measure on the 100 notes of the test set (which we do not have access to) was achieved by the ezDI team with 75.7 [5], using a CRF, BIO scheme, parsing and Brown cluster features, and an SVM to re-build disjoint mentions. The UTH_CCB team was third that year with an F measure of 73.5 on the test set [8].

AS for matching schemes, most works use some form of edit distance (TODO).

3 Related Work

For identification:

- Collobert et al., NLP from scratch, uses a hybrid NN-CRF model with feature embeddings for POS tagging [1]
- Learning character-level representations for part-of-speech tagging does the same thing with embedding of sub-word units [7]
- Conditional random fields as recurrent neural networks formulates CRF inference as recurrent neural networks and back-propagates through the rest of the network [11]
- Non-lexical neural architecture for fine-grained POS Tagging [4]

For mapping:

- Embedding methods for fine grained entity type classification, [9]

4 Model description

4.1 CRF

Our baseline is a CRF with the following features (and combinations thereof):

- word
- lemma
- pos (part of speech)
- normal
- word_length
- prefix
- suffix
- all_caps
- capitalized
- word_pos (word position in the sentence)
- sentence_pos (sentence position in the document)
- sentence_length
- med_prefix (when the word prefix matches a UMLS prefix)
- umls_match_tag_full
- umls_match_tag_prefix
- umls_match_tag_acro

Tagging scheme The standard mention identification methodology consists in tagging tokens with 'B' (first token of the mention), 'I' (other tokens of the mention) or 'O' labels (everything else). This reduces the problem to sequence tagging, which can be solved linearly in the length of the sentence. For example, "the patient suffers from a broken jaw ." will have tags O-O-O-O-O-B-I-O.

About 10% of the mentions we are dealing with are dis-contiguous. While those can technically be represented with a B-I-O scheme, there is a concern that a model such as a CRF will have difficulties propagating information from one part of a mention to another. We remedy this by adding two new labels: 'OD' for tokens which are within the scope of a mention but not part of the mention itself, and 'ID' for tokens which are part of a dis-contiguous mention. For example, "the pain is stongest in the arm ." will have tags O-B-OD-ODOD-OD-ID-O. About 95% of the sentences in our data can be represented with this tagging scheme. It fails however any time two or more mentions overlap. It is worth noting that the vast majority of cases of overlapping mentions in our data are of the form "A and B are C" or "A is B and C". While any per-token tagging scheme which covers all possible configurations would require a prohibitively large tag set, it is easy enough to come up with one which covers all of the aforementioned cases without making our scheme much more complex. The idea is the following: when mentions overlap, we identify each by its "first proper tag". In the example "left arm and shoulder are swollen", the mention "left arm swollen" will be identified by "arm", and the mention "left shoulder swollen" by "shoulder". We introduce a new tag 'In' to denote such an identifying token (the sequence will then be labeled B-In-OD-In-OD-ID). To make the model fully coherent however, we actually need two more tags: 'Bn', which acts as 'In' but for the first word of a mention ("elbow and wrist broken" is labeled Bn-OD-Bn-ID), and 'Ip' for a token which is part of only one of two overlapping mention, but is not the identifying token ("inflamation of left kidney and spleen" will be labeled B-OD-In-Ip-OD-In). This scheme uniquely defines all but 0.5% of our sentences (it still fails in the case of nested mentions, which exist but are quite rare).

We reconstruct mentions from the MAP assignment of tags. The model was trained with the CRF++ package, and achieved an optimal F-measure on the development set (133 notes) of 0.78.

4.2 New tagging models

We implemented two other models using TensorFlow: a neural network which predicts each tag independently(-ish) given the whole sentence (SequNN), and a hybrid neural network / CRF system (NN-CRF, the CRF potentials are the output of a neural network).

Both have a first layer which embeds each token of the sentence into a dimension d space. The embedding of a token is the sum of the embeddings of its features. So the embedding of the token *complaining:* {*lemma: complain, pos: VBG, prefix:com-, suffix, -ing, umls_match_tag_full: O*} is:

$$w_{complaining} = w_{complain} + w_{VBG} + w_{com-} + w_{-ing} + w_O$$

4.2.1 SequNN

I tried adding various neural network architectures on top of the embedding layer, and learning the model with SGD.

Convolutional layer My current network actually consists of a single convolutional layer with window size 5, followed by a ReLU. To be perfectly clear, a convolutional window takes a window $(w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2})$, concatenates them into a dimension $5d$ vector, then applies a linear transformation to dimension d'. A ReLU is then applied, followed by a linear transformation from d' to the number of tags (5 currently), followed by a Soft-Max. (A picture is on its way)

Bi-directional LSTM I also implemented a bi-directional LSTM RNN. I couldn't get it to help however (it did worse than the ConvNet on its own, and putting the ConvNet on top of the RNN layer didn't help).

Prediction targets I then first replaced the marginals of output by the CRF in Section 4.1 by the probabilities output by the NN. However, I could not get past an F-measure of 0.7 with this system.

My next step was then to replace the tag prediction layer by a window prediction layer: instead of giving the probability of a tag, the last layer gives the probability of a tag and its left and right neighbour (125 possibilities). This got me back close to the same performance as the CRF (0.763).

4.2.2 NN-CRF

Model I also trained a neural network to output the potentials of a CRF on the tags. The current architecture is basically the same as for the per-tag (per-window) prediction: embedding layer to window 5 convolution to tanh to pair-wise potentials (5 x 5 matrices for binary potentials, and unary potentials).

Learning The model is trained by SGD on the joint likelihood of the sequence of tags (inference is pretty fast), and back-propagating from the potentials to the NN parameters. The best F-measure obtained is about the same as the vanilla CRF. I'm hoping to play a bit with the structure and learning parameters to improve that.

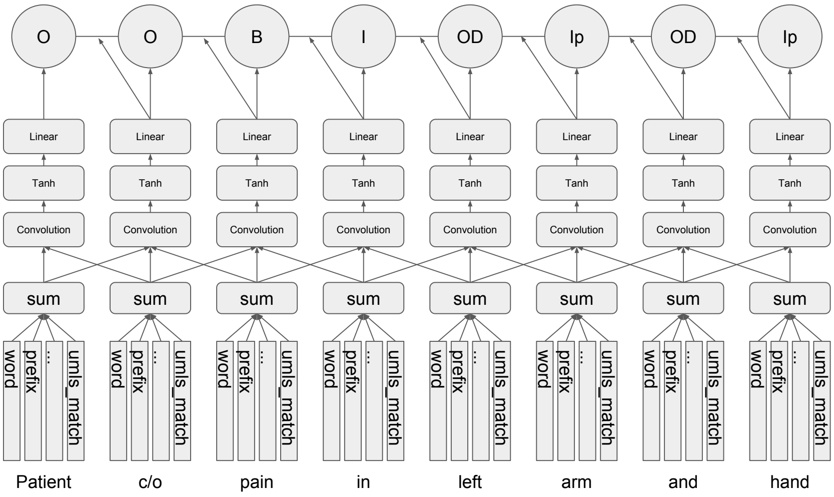

Figure 1: NN-CRF model

4.2.3 Other implementation details

I use the Adam optimizer with a learning rate of 0.02 for NN-CRF and 0.001 for SequNN, and gradient clipping. I have L1 regularization on the word, lemma and normalized word embeddings, and L2 regularization on all the embeddings. No dropout for now.

4.3 Semi-supervised learning

We could add a language modelling objective to get better embeddings.

We could train the first layer(s) (representation and maybe convolution) jointly.

Adding pre-trained word embeddings does not help. I have tried both using them as is, and adding a trainable projection.

References

[1] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

[2] N. Elhadad, S. Pradhan, S. L. Gorman, S. Manandhar, W. Chapman, and G. Savova. Semeval-2015 task 14: Analysis of clinical text. *Notes*, 298(133):100.

[3] G. Glavaš. Takelab: Medical information extraction and linking with mineral.

[4] M. Labeau, K. Löser, A. Allauzen, and R. J. von Neumann. Non-lexical neural architecture for fine-grained pos tagging. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 232–237, 2015.

[5] P. Pathak, P. Patel, V. Panchal, S. Soni, K. Dani, N. Choudhary, and A. Patel. ezdi: A supervised nlp system for clinical narrative analysis. *SemEval-2015*, page 412, 2015.

[6] S. Pradhan, N. Elhadad, W. Chapman, S. Manandhar, and G. Savova. Semeval-2014 task 7: Analysis of clinical text. *SemEval 2014*, 199(99):54, 2014.

[7] C. D. Santos and B. Zadrozny. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1818–1826, 2014.

[8] J. Xu, Y. Zhang, J. Wang, Y. Wu, M. Jiang, E. Soysal, and H. Xu. Uth-ccb: The participation of the semeval 2015 challenge–task 14. *Proceedings of SemEval-2015*, 2015.

[9] D. Yogatama, D. Gillick, and N. Lazic. Embedding methods for fine grained entity type classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International*

Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL, pages 26–31, 2015.

[10] Y. Zhang, J. Wang, B. Tang, Y. Wu, M. Jiang, Y. Chen, and H. Xu. Uth_ccb: A report for semeval 2014–task 7 analysis of clinical text. *SemEval 2014*, page 802, 2014.

[11] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional random fields as recurrent neural networks. *arXiv preprint arXiv:1502.03240*, 2015.