A Bootstrapping Approach for Multi-Relation Extraction

Jon Elsas & Jaime Arguello

We propose a framework for learning a set of relations concurrently. The relations are set *a priori*. The system does the rest. This work expands on our experience with single-relationship extraction during the first half of the semester.

Single-Relation Extraction

Our algorithm for single-relation extraction attempts to find entity pairs *(A,B)* that are involved in relationship *R*. It bootstraps from learned *entity pairs (A,B)* and *contexts (S)*.

The probability that entity pair $(A,B)_i$ is in relation R is give by:

$$\textbf{(1)} \qquad P(+R \mid (A,B)_i) = \sum_j \left[P(+R \mid S_j) \times P(S_j \mid (A,B)_i) \right]$$

The probability that context S_i marks a relation R is given by:

$$\textbf{(2)} \qquad P(+R \mid S_i) = \sum_j \left[P(+R \mid (A,B)_j) \times P((A,B)_j \mid S_i) \right]$$

Problem:
One major problem with *single-relation extraction* is that often times entity pair $(A,B)_i$ does not always occur in a context S_j that exclusively marks relation R.

Consider trying to find entities that are in a causal relation. We may find "*Studies show that smoking causes cancer"* and *"This article talks about smoking and cancer"*. The context "_____ *causes* _____" exclusively marks causal relations. The context "_____*and* _____" is not exclusively used to mark causal relations (e.g. "Tom and Jerry, "Apples and Oranges", "Up and down", etc.). Our single-relation extractor is not robust against overly general contexts and so precision decreases exponentially (or so it seems).

We hope to mitigate this problem by extending our model to multiple-relation extraction.

Multi-Relation Extraction

We will concurrently learn entities for a set of relations R. Each entity pair (A,B) will be represented as a vector of probabilities $\vec{P}_e$ where $\left|\vec{P}_e\right| = |R|$ and the nth element in $\vec{P}_e = P(+R_n \mid (A,B))$. Likewise, each context S will be represented as a vector of probabilities $\vec{P}_s$ where $\left|\vec{P}_s\right| = |R|$ and the nth element in $\vec{P}_s = P(+R_n \mid S)$. Thus, formulas (1) and (2) above become:

(1'') $$P(+R_n \mid (a,b)_i) = \frac{\sum_j \left[P(+R_n \mid s_j) \times P(s_j \mid (a,b)_i)\right]}{\sum_k \sum_j \left[P(+R_k \mid s_j) \times P(s_j \mid (a,b)_i)\right]}$$

(2'') $$P(+R_n \mid s_i) = \frac{\sum_j \left[P(+R_n \mid (a,b)_j) \times P((a,b)_j \mid s_i)\right]}{\sum_k \sum_j \left[P(+R_k \mid (a,b)_j) \times P((a,b)_j \mid s_i)\right]}$$

Tentative Solution 1 – Limit Scope

One concern with this approach is that we are implicitly making the assumption that our set of relations $\{R_1, \ldots R_n\}$ represent a complete set of possible relations over any pair of learned entities (a,b). Although this is a strong assumption, we believe it will not cause a major problem with the algorithm and is logically consistent with many choices of relations to learn.

For instance, in the medical domain we may attempt to learn the three relations TreatmentFor(*medication*, *disease*), SideEffectOf(*side-effect*, *medication*) and SymptomOf(*symptom*, *disease*). For a pair of entities sampled from a set of *diseases*, *symptoms* and *side-effects*, the stated relationships are the only logical connection between any (*medication*, *side-effect*), (*disease*, *medication*) or (*disease*, *symptom*) pair. Thus, we would constrain the system to only consider local contexts (e.g. sentences) that contain one of these three entity-type pairs.

Some complications are introduced when many of the relationships being learned can exist between a pair of entities, such as FatherOf and MotherOf over the set of pairs of person entities.

Tentative Solution 2 – Proxy Catch-All Relation

Alternatively, we could also ensure complete coverage over all relations by introducing a proxy "catch-all" relation. For instance, if we wish to learn relations R_1, R_2, and R_3, we would also introduce relation R_∞, which would logically map to $\overline{(R_1 \vee R_2 \vee R_3)}$. R_1, R_2, R_3, and R_∞ describe ALL possible relations over the space of entities we're interested in.

If we limit our search to relations that are asynchronous or directional (Cause → Effect, Medicine → Side-effect, Illness → Symptom, etc.), we could learn entities for R_∞ by reversing the directionality of our learned entities for relations $\{R_1, R_2 \ldots R_{\infty-1}\}$. As usual, we would use these entities to learn more contexts that mark R_∞, and so forth.

Literature Review

This work is primarily building off of the work of Rosie Jones (see her dissertation, 2005) and Ghani & Nigam (*Analyzing the Effectiveness and Applicability of Co-training* in CIKM 2000). Other papers we will be looking at include:

M. Collins, Yoram Singer, *Unsupervised Models for Named Entity Classification*. in: EMNLP 99, 1999.

Neel Sundaresan, Jeonghee Yi, *Mining the Web for Relations* in Proceedings of the Ninth International World Wide Web Conference (WWW09), 2000. http://www9.org/w9cdrom/363/363.html

Cheng Niu, Wei Li, Jihong Ding, Rohini K. Srihari *A Bootstrapping Approach to Named Entity Classification Using Successive Learners* in ACL 2003. http://acl.ldc.upenn.edu/acl2003/main/pdfs/Niu.pdf

Takaaki Hasegawa, Satoshi Sekine and Ralph Grishman. *Discovering Relations among Named Entities from Large Corpora* in ACL 2005. http://acl.ldc.upenn.edu/acl2004/main/pdf/195_pdf_2-col.pdf

Yunbo Cao, Hang Li and Li Lian, *Uncertainty Reduction in Collaborative Bootstrapping: Measure and Algorithm* in ACL 2005 http://acl.ldc.upenn.edu/acl2003/main/pdf/Cao.pdf

Roman Yangarber, *Counter-Training in Discovery of Semantic Patterns* in ACL 2005 http://acl.ldc.upenn.edu/acl2003/main/pdf/Yangarber.pdf

Dependencies on other components:

Although our system should be able to operate on raw, un-annotated text, the addition of noun phrase annotations as well as some level of named entity annotation is likely to boost performance. Ideally, the named entity tagging would focus on to the types of relations being learned. For example, if we are learning the relations described above, tagging of diseases, medications, etc. should boost precision considerably.