On Negative Examples for Distantly-supervised Relation Extraction

Mihai Surdeanu
University of Arizona
msurdeanu@email.arizona.edu

November 26, 2012

1 Discussion

A careful reader of our recent EMNLP paper (Surdeanu et al., 2012) will observe a somewhat untraditional notation: we use P_i to denote the known positive labels for an entity tuple i, *and N_i to denote the set of known negative labels for the same tuple*, that is, labels for which tuple i serves as a negative example (introduced in Section 4). This is uncommon: traditionally at training time, an entity tuple (e_1, e_2) that does not exist in the training DB is considered a negative example for all possible labels, which makes the maintenance of an explicit set N_i unnecessary. Most previous work on this topic, including our EMNLP 2012 paper, used this heuristic.

However, in the context of the KBP slot filling task, where most infoboxes provided as training data are incomplete, this heuristic is not ideal. For example, let's assume that we have an incomplete infobox for *Rachmaninoff* with a single slot (*person:country_of_birth*, *Russia*), and during training we see the tuple (*Rachmaninoff*, *United States*). What label should we assign to this tuple? According to the above heuristic, this tuple is a negative example for all the valid labels for `PERSON` entities, including *person_country_of_death*. But this is wrong: in fact, Rachmaninoff died in the United States. We just did not have this information in the corresponding infobox.

A better heuristics, which, to my knowledge, was discovered at the same time by (Sun et al., 2011) and (Surdeanu et al., 2011), is to consider the tuple a negative example only for the labels which exist in e_i's infobox with a different label. More formally, using our notation, N_i for the ith tuple (e_1, e_2) is defined as: $\{r_j \mid r_j(e_1, e_k) \in \mathcal{D}, e_k \neq e_2, r_j \notin P_i\}$. That is, for the above example, (*Rachmaninoff*, *United States*) should be considered as negative example *only* for *person:country_of_birth* because this is the only thing we know with certainty about *Rachmaninoff* in our training dataset.

Modeling this heuristic using local, one-vs-rest classifiers is trivial, as illustrated by both (Sun et al., 2011) and (Surdeanu et al., 2011): you just create different negative example sets for each label, according to the data available in the infoboxes. However, when dealing with a joint model, implementing this is not exactly trivial.[1] To pat ourselves on the back, we designed our EMNLP algorithm from the very beginning under the assumption that one needs to maintain N_i explicitly, although we left the empirical analysis of the heuristic discussed here as future work. So our algorithm works as is with this new heuristic (you just create N_i differently).

2 Modified Hoffmann Algorithm

Another relation extraction (RE) algorithm that I really like was proposed by (Hoffmann et al., 2011). This algorithm is beautifully simple yet it performs very well. So it is worth exploring how to adapt it to the the heuristic presented above.

Algorithm 1 shows the modified algorithm. I followed the same notations as the original paper, with only a couple of significant changes: in Algorithm 1, $\mathbf{y}_\mathbf{i}^+$ is equivalent with $\mathbf{y_i}$ in (Hoffmann et al., 2011), that is, it stores the set of valid labels for tuple i. This

[1] But not too hard either. Otherwise, these notes would be a conference paper rather than some informal musings.

Definitions:
Same inputs and definitions as (Hoffmann et al., 2011), with two new elements:
(a) $\mathbf{y_i^+} = \mathbf{relVector}(e_j, e_k)$ is similar to $\mathbf{y_i}$ from (Hoffmann et al., 2011);
(b) $\mathbf{y_i^-}$ is the corresponding vector for negative examples (e_j, e_k), with bit r set to 1 if it is known that the tuple (e_j, e_k) cannot have label r.
Notations:
$\bigcap$, $\bigcup$, $\setminus$: vector bit AND, OR and DIFF
Computation:
$\mathbf{\Theta} \leftarrow \mathbf{0}$
for $t = 1$ **to** T **do**
  for $i = 1$ **to** n **do**
    $(\mathbf{y}', \mathbf{z}') \leftarrow \arg\max_{\mathbf{y},\mathbf{z}} p(\mathbf{y}, \mathbf{z} | \mathbf{x_i}; \theta)$
    if $\mathbf{y}' \bigcap \mathbf{y_i^+} \neq \mathbf{y_i^+}$ *or* $\mathbf{y}' \bigcap \mathbf{y_i^-} \neq \mathbf{0}$ **then**
      $\mathbf{y^{update}} \leftarrow (\mathbf{y}' \bigcup \mathbf{y_i^+}) \setminus (\mathbf{y}' \bigcap \mathbf{y_i^-})$
      $\mathbf{z}^* \leftarrow \arg\max_{\mathbf{z}} p(\mathbf{z} | \mathbf{x_i}, \mathbf{y^{update}}; \theta)$
      $\mathbf{\Theta} \leftarrow \mathbf{\Theta} + \theta(\mathbf{x_i}, \mathbf{z}^*) - \theta(\mathbf{x_i}, \mathbf{z}')$
      // optionally, soft updates for unknown labels here
    else
      // optionally, soft updates for unknown labels here

Algorithm 1: The Hoffmann algorithm adapted to work with information from incomplete infoboxes.

is equivalent to P_i in our EMNLP paper. I added $\mathbf{y_i^-}$, to store the set of labels for which tuple i serves as a negative example. This is equivalent to N_i above and, as discussed, this is created from the partial infoboxes.

The first thing that changes is the update condition (the first `if` condition). Under the new heuristic, the update condition is true if a positive label is missed ($\mathbf{y}' \bigcap \mathbf{y_i^+} \neq \mathbf{y_i^+}$) or a negative one is predicted ($\mathbf{y}' \bigcap \mathbf{y_i^-} \neq \mathbf{0}$). Note that this condition does not say anything about unknown labels, that is, labels that are not either in $\mathbf{y_i^+}$ or $\mathbf{y_i^-}$. The model can generate such labels with no effect on the update. Since we do not know what to do with such labels (they may or may not be correct), we let them fly.

If the update is triggered, $\mathbf{y^{update}}$ is created from the set of labels currently predicted for tuple i, to which we append any known positives missed and remove any negative labels that are predicted. Again, this means that we do not touch the unknown labels that are predicted. The rest of the update process if similar to the original Hoffmann algorithm. I, however, added two optional steps (as comments in Algorithm 1), which indicate that it is possible to penalize somewhat (with a "soft" update) the unknown labels that are predicted by the model. For example, one could add a hyper parameter (between 0 and 1) to indicate the strength of the "soft" update (closer to 0 indicates a weak update; closer to 1 a strong one), and do a negative update on $\mathbf{\Theta}$ using the current datum's weights multiplied by this hyper parameter. This way, one could penalize the prediction of unknown labels, but with a penalty smaller than the penalty used for labels that are known to be incorrect.

3 Evaluation

Results are hopefully coming soon (although I make no promises).

One thing I would like to emphasize on this issue, is that an empirical comparison between algorithms using this new heuristic versus the standard one is a little tricky. For example, it is incorrect to compare the P/R/F1 scores for an algorithm that uses all negative examples generated by the traditional heuristic against the same algorithm that uses the fewer

negative examples generated by the new heuristic[2]. Simply because the training data is different, one algorithm may obtain a different balance between precision and recall, which would give the illusion of a better F1. The only way to compare these algorithms would be to plot the entire P/R curves, where for each algorithm we plot the various P/R scores obtained using different subsampling rates for negative examples.

References

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Ang Sun, Ralph Grishman, Wei Xu, and Bonan Min. 2011. New York University 2011 system for KBP slot filling. In *Proceedings of the Text Analytics Conference*.

Mihai Surdeanu, Sonal Gupta, John Bauer, David McClosky, Angel X. Chang, Valentin I. Spitkovsky, and Christopher D. Manning. 2011. Stanford's distantly-supervised slot-filling system. In *Proceedings of the Text Analytics Conference*.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing and Natural Language Learning (EMNLP-CoNLL)*.

[2]The new heuristic generates fewer negative examples because it does not perform negative updates on the unknown labels.