A High Accuracy Method for Semi-supervised Information Extraction

Stephen Tratz
Pacific Northwest National Laboratory
Richland, WA 99352
stephen.tratz@pnl.gov

Antonio Sanfilippo
Pacific Northwest National Laboratory
Richland, WA 99352
antonio.sanfilippo@pnl.gov

Abstract

Customization to specific domains of discourse and/or user requirements is one of the greatest challenges for today's Information Extraction (IE) systems. While demonstrably effective, both rule-based and supervised machine learning approaches to IE customization pose too high a burden on the user. Semi-supervised learning approaches may in principle offer a more resource effective solution but are still insufficiently accurate to grant realistic application. We demonstrate that this limitation can be overcome by integrating fully-supervised learning techniques within a semi-supervised IE approach, without increasing resource requirements.

1 Introduction

Customization to specific discourse domains and/or user requirements is one of the greatest challenges for today's Information Extraction (IE) systems. While demonstrably effective, both rule-based and supervised machine learning approaches to IE customization require a substantial development effort. For example, Aone and Ramos-Santacruz (2000) present a rule-based IE system which handles 100 types of relations and events. Building such a system requires the manual construction of numerous extraction patterns supported by customized ontologies. Soderland (1999) uses supervised learning to induce a set of rules from hand-tagged training examples. While Sonderland suggests that the human effort can be reduced by interleaving learning and manual annotation activities, the creation of training data remains an onerous task.

To reduce the knowledge engineering burden on the user in constructing and porting an IE system, unsupervised learning has been utilized, e.g. Riloff (1996), Yangarber et al. (2000), and Sekine (2006). Banko et al. (2007) present a self-supervised system that aims to avoid the manual IE customization problem by extracting all possible relations of interest from text. Stevenson and Greenwood (2005) propose a weakly supervised approach to *sentence filtering* that uses semantic similarity and bootstrapping to acquire IE patterns. Stevenson's and Greenwood's approach provides some of the best available results in weakly supervised IE to date, with 0.58 F-measure. While very good, an F-measure of 0.58 does not provide sufficient reliability to grant use in a production system.

In this paper, we show that it is possible to provide a significant improvement over Stevenson's and Greenwood's results, without increasing resource requirements, by integrating fully-supervised learning techniques within a weakly supervised IE approach.

1.1 Learning Algorithm

Our method is modeled on the approach developed by Stevenson and Greenwood (2005) but uses a different technique for ranking candidate patterns. Stevenson's and Greenwood's algorithm takes as data inputs a small set of initial seed patterns and a corpus of documents, and uses any of several semantic similarity measures (Resnik, 1995; Jiang and Conrath, 1997; Patwardhan et al., 2003) to iteratively identify patterns in the document corpus

Proceedings of NAACL HLT 2007, Companion Volume, pages 169–172,
Rochester, NY, April 2007.

that bear a strong resemblance to the seed patterns. After each iteration, the top-ranking candidate patterns are added to the seed patterns and removed from the corpus. Our approach differs from that of Stevenson and Greenwood in that we use a supervised classifier to rank candidate patterns. This grants our system greater robustness and flexibility because the weight of classification features can be automatically determined within a supervised classification approach.

In building supervised classifiers to rank candidate patterns at each iteration, we use both positive and negative training examples. Instead of creating manually annotated training examples, we follow an active learning approach where training examples are automatically chosen by ranking candidate patterns in terms of cosine similarity with the seed patterns. More specifically, we select patterns that have the lowest similarity with seed patterns to be the negative training examples. We hypothesized that these negative examples would contain many of the uninformative features occurring throughout the corpus and that using these examples would enable the classifier to determine that these features would not be useful.

The pattern learning approach we propose includes the following steps.

1. An unannotated corpus is required as input. For each sentence, a set of features is extracted. This information becomes S_{cand}, the set of all candidate patterns.
2. The user defines a set of seed patterns, S_{seed}. These patterns contain features expected to be found in a relevant sentence.
3. The cosine measure is used to determine the distance between the patterns in S_{seed} and S_{cand}. The patterns in S_{cand} are then ordered by their lowest distance to a member of S_{seed}.
4. The α highest ranked patterns in S_{cand} are added to S_{pos}, the set of positive training examples.
5. S_{seed} and S_{acc} are added to S_{pos}. S_{neg}, the set of negative training examples is constructed from β+iteration*γ of the lowest ranked patterns in S_{cand}. Then, a maximum entropy classifier is built using S_{pos} and S_{neg} as training data.
6. The classifier is used to score each pattern in S_{cand}. S_{cand} is then sorted by these scores.
7. The top δ patterns in S_{cand} are added to S_{acc} and removed from S_{cand}.
8. If a suitable stopping point has been reached, the process ends. Otherwise, S_{pos} and S_{neg} are emptied and the process continues at step 6.

We set α to 5, β to 20, γ to 15, δ to 5, and used the following linguistic processing tools: (1) the OpenNLP library (opennlp.sourceforge.net) for sentence splitting and named-entity recognition, and (2) Connexor for syntactic parsing (Tapanainen and Järvinen, 1997). For the classifier, we used the OpenNLP MaxEnt implementation (maxent.sourceforge.net) of the maximum entropy classification algorithm (Berger et al. 1996). We used the MUC-6 data set as the testing ground for our proposed approach.

1.2 Description of Features Used

Stevenson and Greenwood (2005) use subject-verb-object triples for their features. We use a richer feature set. Our system can easily accommodate more features because we let the maximum entropy classifier determine the weight for the features. Stevenson's and Greenwood's approach determines weights using semantic similarity and would require significant changes to take into account various other features, especially those for which a WordNet (Fellbaum, 1998) similarity score is not available.

We use single tokens, token combinations, and semantic information to inform our IE pattern extraction system. Lexical items marked by the named-entity recognition system as PERSON or ORGANIZATION are replaced with 'person' and 'organization', respectively. Number tokens are replaced with 'numeric'. Single Token Features include:

- All words in the sentence and all hypernyms of the first sense of the word with attached part-of-speech
- All words in the sentence with attached dependency
- The verb base of each nominalization and the verb's first sense hypernyms are included.

Token Combinations include:

- All bigrams from the sentence
- All subject-object pairs
- All parent-child pairs from the parse tree

- A specially marked copy of the parent-child pairs where the main verb is the parent.

We also added semantic features indicating if a PERSON or ORGANIZATION was detected within the sentence boundaries. Table1 provides an example where a simple sentence is mapped into the set of features we have just described.

```
Alan G. Spoon, 42, will succeed
Mr. Graham as president of the
company.
```

Single Token Features
With attached dependencies: attr:person, subj:person, mod:numeric, v-ch:will, main:succeed, obj:person, copred:as, pcomp:president, mod:of, det:the, pcomp:company *With part-of-speech tags:* n:person, v:succeed, v:will, dt:the, n:company, n:institution, n:social_group, n:group, n:organization, n:person, n:president, n:executive, n:corporate_executive, n:administrator, n:head, n:leader, n:organism, n:living_thing, n:object, n:entity, num:numeric, abbr:person, prp:as, prp:of, v:control, v:declare, v:decree, v:express, v:ordain, v:preside, v:state
Token Combinations
Bigrams: person+comma, comma+numeric, numeric+comma, comma+will, will+succeed, succeed+person, person+as, as+president, president+of, of+the, the+company *Subject Object Pairs:* sop:person+person *Parent-Child Pairs:* pc:person+person, pc:person+numeric, pc:will+person, pc:succeed+will, pc:succeed+person, pc:succeed+as, pc:as+president, pc:president+of, pc:of+company, pc:company+the *Main Verb Parent-Child Pairs:* mvpc:succeed+person, mvpc:succeed+will, mvpc:succeed+as
Semantic Features
hasOrganization, hasPerson

Table 1: Feature representation of a simple sentence.

The seeds we used are adapted from the seed patterns employed by Stevenson and Greenwood. As shown in Table 2, only a subset of the features described above is used in the seed patterns.

2 Evaluation

We used the document collection which was initially developed for the Sixth Message Understanding Conference (MUC-6) as ground truth data set to evaluate our approach. The MUC-6 corpus (www.ldc.upenn.edu) is composed of 100 Wall Street Journal documents written during 1993 and 1994. Our task was to detect sentences which included management succession patterns, such as those shown in Table 2.

1: subj:organization, main:appoint, obj:person, hasPerson, hasOrganization
2: subj:organization, main:elect, obj:person, hasOrganization, hasPerson
3: subj:organization, main:promote, obj:person, hasOrganization, hasPerson
4: subj:organization, main:name, obj:person, hasOrganization, hasPerson
5: subj:person, main:resign, hasPerson
6: subj:person, main:depart, hasPerson
7: subj:person, main:quit, hasPerson

Table 2: Feature representation of seed patterns.

The version of the MUC-6 corpus produced by Soderland (1999) provided us with a specification of succession patterns at the sentence level, but as shown in Table 3 did not include the source text. We reconstructed the original text by automatically aligning the succession patterns in the sentence structures in Soderland's version of the MUC-6 corpus with the sentences in the original MUC-6 corpus. This alignment produced a set of 1581 sentences, of which 134 contained succession patterns.

```
@S[
  {SUBJ  @CN[ FOX ]CN }
  {VB  NAMED @NAM }
  {OBJ  @PN[ LUCILLE S. SALHANY ]PN , @PS[
CHAIRMAN ]PS OF @CN[ FOX INC. ]CN 'S
TELEVISION PRODUCTION ARM , }
  {REL_V  TO SUCCEED @SUCCEED HIM . }
]@S 9301060123-5
@@TAGS Succession {PersonIn @PN[ LUCILLE S.
SALHANY ]PN}+ {Post @PS[ CHAIRMAN ]PS}+
{Org @CN[ FOX INC. ]CN}_ @@COVERED_BY
@@ENDTAGS
```

Table 3: Data sample from Soderland test set.

As shown in Figure 1, our best score of 0.688 F-measure was obtained on the 36th iteration; at the end of this iteration, our algorithm selected 180 sentences including 108 of the sentences that contained succession patterns. This is a significant improvement over the 0.58 F-measure score

reported by Stevenson and Greenwood (2005) for the same task. The use of a supervised classification approach to the ranking of candidate patterns with a richer feature set were the two determinant factors in achieving such improvement.

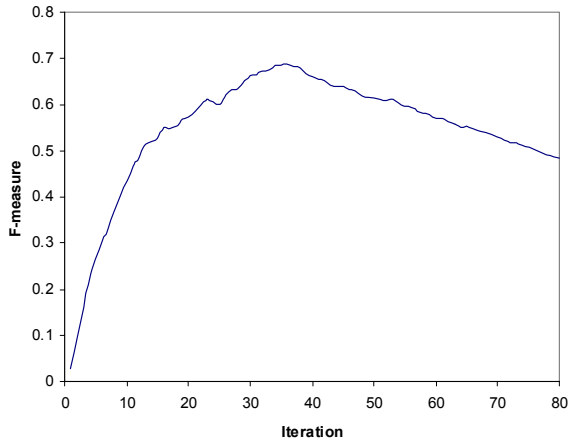

Figure 1: Evaluation results with MUC-6 data.

3 Conclusions

Our results show a substantial improvement over previous efforts in weakly supervised IE methods, suggesting that weakly supervised methods can be made to rival rule-based or fully supervised approaches both in resource effectiveness and accuracy. We plan to verify the strength of our approach evaluating against other ground truth data sets. We also plan to detail how the various features in our classification model contribute to ranking of candidate patterns. An additional area of envisioned improvement regards the use of a random sub selection of negative candidate patterns as training samples to counteract the presence of sentence fragments among low-ranking candidate patterns. Finally, we intend to evaluate the benefit of having a human in the loop in the first few iterations to filter out patterns chosen by the system.

References

C. Aone and M. Ramos-Santacruz. 2000. REES: A Large-Scale Relation and Event Extraction System, pages 76-83, In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP 2000)*, Seattle.

M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. 2007. Open Information Extraction from the Web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*. Hyderabad, India.

A. Berger, S. Della Pietra and V. Della Pietra (1996) A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics,* volume 22, number 1, pages 39-71.

C. Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database and some of its Applications*. MIT Press, Cambridge, MA.

J. Jiang and D. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. *In Proceedings of the 10th International Conference on Research in Computational Linguistics*, Taiwan.

S. Patwardhan, S. Banerjee, and T. Pederson. 2003. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the 4th International Conferences on Intelligent Text Processing and Computational Linguistics*, pages 241-257, Mexico City.

P. Resnik. 1995. Using Information Content to evaluate Semantic Similarity in a Taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 448-452, Montreal, Canada.

E. Riloff and R. Jones. 1999. Learning Dictionaries for Information Extraction by Multi-level Bootstrapping. In *Proceedings of the 16th National Conference on Artificial Intelligence*. Orlando, Florida.

S. Sekine. 2006. On-Demand Information Extraction. In Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions. Sydney, Australia.

S. Soderland. 1999. Learning Information Extraction Rules for Semi-structured and free text. *Machine Learning*, 31(1-3):233-272.

M. Stevenson and M. A. Greenwood. 2005. A Semantic Approach to IE Pattern Induction. In *Proceedings of the 43rd Annual Meeting of the ACL (ACL 05)*, Ann Arbor, Michigan.

P. Tapanainen and Timo Järvinen. 1997. A non-projective dependency parser. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, pages 64–71, Washington D.C. Association for Computational Linguistics.

R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen. 2000. Automatic acquisition of domain knowledge for information extraction. In *Proceedings of the 18th International Conference of Computational Linguistics (COLING 2002)*, Taipei.