Semantic Enrichment of Text with Background Knowledge

Anselmo Peñas
UNED NLP & IR Group
Juan del Rosal, 16
28040 Madrid, Spain
anselmo@lsi.uned.es

Eduard Hovy
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292-6695
hovy@isi.edu

Abstract

Texts are replete with gaps, information omitted since authors assume a certain amount of background knowledge. We describe the kind of information (the formalism and methods to derive the content) useful for automated filling of such gaps. We describe a stepwise procedure with a detailed example.

1 Introduction

Automated understanding of connected text remains an unsolved challenge in NLP. In contrast to systems that harvest information from large collections of text, or that extract only certain prespecified kinds of information from single texts, the task of extracting and integrating all information from a single text, and building a coherent and relatively complete representation of its full content, is still beyond current capabilities.

A significant obstacle is the fact that text always omits information that is important, but that people recover effortlessly. Authors leave out information that they assume is known to their readers, since its inclusion (under the Gricean maxim of minimality) would carry an additional, often pragmatic, import. The problem is that systems cannot perform the recovery since they lack the requisite background knowledge and inferential machinery to use it.

In this research we address the problem of automatically recovering such omitted information to 'plug the gaps' in text. To do so, we describe the background knowledge required as well as a procedure for recognizing where gaps exist and determining which kinds of background knowledge are needed.

We are looking for the synchronization between the text representation achievable by current NLP and a knowledge representation (KR) scheme that can permit further inference for text interpretation.

1.1 Vision

Clearly, producing a rich text interpretation requires both NLP and KR capabilities. The strategy we explore is the enablement of bidirectional communication between the two sides from the very beginning of the text processing. We assume that the KR system doesn't require a full representation of the text meaning, but can work with a partial interpretation, namely of the material explicitly present in the text, and can then flesh out this interpretation as required for its specific task. Although the NLP system initially provides simpler representations (even possibly ambiguous or wrong ones), the final result contains the semantics of the text according to the working domain.

In this model, the following questions arise: How much can we simplify our initial text representation and still permit the attachment of background knowledge for further inference and interpretation? How should background knowledge be represented for use by the KR system? How can the incompleteness and brittleness typical of background knowledge (its representational inflexibility, or limitation to a single viewpoint or expressive phrasing) (Barker 2007) be overcome? In what sequence can a KR system enrich an initial and/or impoverished reading, and how can the enrichment benefit subsequent text processing?

1.2 Approach

Although we are working toward it, we do not yet have such a system. The aim of our current work is to rapidly assemble some necessary pieces and explore how to (i) attach background knowledge to flesh out a simple text representation and (ii) there by make explicit the meanings attached to some of its syntactic relations. We begin with an initial simple text representation, a background knowledge base corresponding to the text, and a simple formalized procedure to attach elements from the background knowledge to the entities and implicit relations present in the initial text representation.

Surprisingly, we find that some quite simple processing can be effective if we are able to contextualize the text under interpretation.

For our exploratory experiments, we are working with a collection of 30,000 documents in the domain of US football. We parsed the collection using a standard dependency parser (Marneffe and Manning, 2008; Klein and Maning, 2003) and, after collapsing some syntactic dependencies, obtained the simple textual representations shown in Section 2. From them, we built a Background Knowledge Base by automatically harvesting propositions expressed in the collection (Section 3). Their frequency in the collection lead the enrichment process: given a new text in the same domain, we build exactly the same kind of representation, and attach the background knowledge propositions as related to the text (Section 4).

Since this is an exploratory sketch, we cannot provide a quantitative evaluation yet, but the qualitative study over some examples suggest that this simple framework is promising enough to start a long term research (Section 5). Finally, we conclude with the next steps we want to follow and the kind of evaluation we plan to do.

2 Text Representation

The starting text representation must capture the first shot of what's going on in the text, taking some excerpts into account and (unfortunately) losing others. After the first shot, in accord with the purpose of the reading, we will "contextualize" each sentence, expanding its initial representation with the relevant related background knowledge in our base.

During this process of making explicit the implicit semantic relations (which we call contextualization or interpretation) it will become apparent whether we need to recover some of the discarded elements, whether we need to expand some others, etc. So the process of interpretation is identified with the growing of the context (according to the KB) until the interpretation is possible. This is related to some well-known theories such as the Theory of Relevance (Sperber and Wilson, 1995). The particular method we envisage is related to Interpretation as Abduction (Hobbs et al. 1993).

How can the initial information be represented so as to enable the context to grow into an interpretation? We hypothesize that:

1. Behind certain syntactic dependencies there are semantic relations.
2. In the case of dependencies between nouns, this semantic relation can be made more explicit using verbs and/or prepositions. The knowledge base must help us find them.

We look for a semantic representation close enough to the syntactic representation we can obtain from the dependency graph. The main syntactic dependencies we want to represent in order to enable enrichment are:

1. Dependencies between nouns such as noun-noun compounds (nn) or possessive (poss).
2. Dependencies between nouns and verbs, such as subject and object relations.
3. Prepositions having two nouns as arguments. Then the preposition becomes the label for the relation between the two nouns, being the object of the preposition the target of the relation.

For these selected elements, we produce two very simple transformations of the syntactic dependency graph:

1. Invert the direction of the syntactic dependency for the modifiers. Since we work with the hypothesis that behind a syntactic dependency there is a semantic relation, we record the direction of the semantic relation.
2. Collapse the syntactic dependencies between verb, subject, and object into a single semantic relation. Since we are assuming that the verb is the more explicit expression of a semantic relation, we fix this in the initial representation. The subject will be the source of the relation and the object will be the target of the relation. When the verb has more arguments we consider its expansion as a new node as referred in Section 4.4.

Figure 1 shows the initial minimal representation for the sentence we will use for our discussion:

```
San_Francisco's Eric_Davis intercepted a Steve_Walsh pass on the next series to set_up a seven-yard Young touchdown pass to Brent_Jones.
```

Notice that some pieces of the text are lost in the initial representation of the text as for example "`on the next series`" or "`seven-yard`".

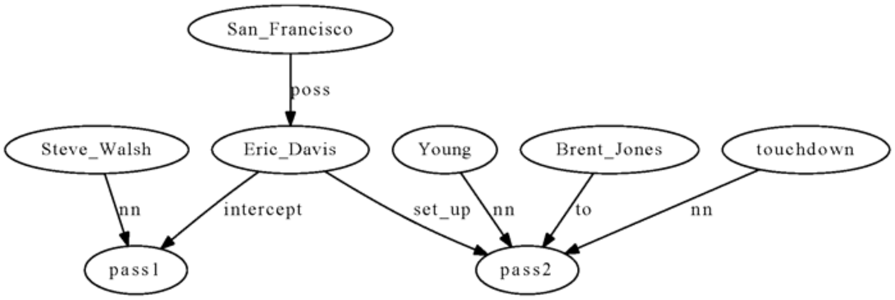

Figure 1. Representation of the sentence: `San_Francisco's Eric_Davis intercepted a Steve_Walsh pass on the next series to set_up a seven-yard Young touchdown pass to Brent_Jones.`

3 Background Knowledge Base

The Background Knowledge Base (BKB) is built from a collection in the domain of the texts we want to semanticize. The collection consists of 30,826 New York Times news about American football, similar to the kind of texts we want to interpret. The elements in the BKB (3,022,305 in total) are obtained as a result of applying general patterns over dependency trees. We take advantage of the typed dependencies (Marneffe and Manning, 2008) produced by the Stanford parser (Klein and Maning, 2003).

3.1 Types of elements in the BKB

We distinguish three elements in our Background Knowledge Base: Entities, Propositions, and Lexical relations. All of them have associated their frequency in the reference collection.

Entities

We distinguish between entity classes and entity instances:

1. Entity classes: Entity classes are denoted by the nouns that participate in a copulative relation or as noun modifier. In addition, we introduce two special classes: Person and Group. These two classes are related to the use of pronouns in text. Pronouns "I", "he" and "she" are linked to class Person. Pronouns "we" and "they" are linked to class Group. For example, the occurrence of the pronoun "he" in "He threw a pass" would produce an additional count of the proposition "person:throw:pass".
2. Entity Instances: Entity instances are indicated by proper nouns. Proper nouns are identified by the part of speech tagging. Some of these instances will participate in the "has-instance" relation (see below). When they participate in a proposition they produce proposition instances.

Propositions

Following Clark and Harrison (2009) we call *propositions* the tuples of words that have some determined pattern of syntactic relations among them. We focus on NVN, NVNPN and NPN proposition types. For example, a NVNPN proposition is a full instantiation of:

`Subject:Verb:Object:Prep:Complement`

The first three elements are the subject, the verb and the direct object. Fourth is the preposition that attaches the PP complement to the verb. For simplicity, indirect objects are considered as a Complement with the preposition "to".

The following are the most frequent NVN propositions in the BKB ordered by frequency.

NVN 2322 'NNP':'beat':'NNP'
NVN 2231 'NNP':'catch':'pass'
NVN 2093 'NNP':'throw':'pass'
NVN 1799 'NNP':'score':'touchdown'
NVN 1792 'NNP':'lead':'NNP'
NVN 1571 'NNP':'play':'NNP'
NVN 1534 'NNP':'win':'game'
NVN 1355 'NNP':'coach':'NNP'
NVN 1330 'NNP':'replace':'NNP'
NVN 1322 'NNP':'kick':'goal'
NVN 1195 'NNP':'win':'NNP'
NVN 1155 'NNP':'defeat':'NNP'
NVN 1103 'NNP':'gain':'yard'

The 'NNP' tag replaces specific proper nouns found in the proposition.

When a sentence has more than one complement, a new occurrence is counted for each complement. For example, given the sentence `"Steve_Walsh threw a pass to Brent_Jones in the first quarter"`, we would add a count to each of the following propositions:

```
Steve_Walsh:throw:pass
Steve_Walsh:throw:pass:to:Brent_Jones
Steve_Walsh:throw:pass:in:quarter
```

Notice that right now we include only the heads of the noun phrases in the propositions.

We call *proposition classes* the propositions that only involve instance classes (e.g., “`person:throw:pass`”), and *proposition instances* those that involve at least one entity instance (e.g., “`Steve_Walsh:throw:pass`”).

Proposition instances are useful for the tracking of a entity instance. For example, “`'Steve_Walsh':'supplant':'John_Fourcade':'as':'quarterback'`”. When a proposition instance is found, it is stored also as a proposition class replacing the proper nouns by a special word (NNP) to indicate the presence of a entity instance.

The enrichment of the text is based on the use of most frequent proposition classes.

Lexical relations

At the moment, we make use of the copulative verbs (detected by the Stanford’s parser) in order to extract “is”, and “has-instance” relations:

1. Is: between two entity classes. They denote a kind of identity between both entity classes, but not in any specific hierarchical relation such as hyponymy. Neither is a relation of synonymy. As a result, is somehow a kind of underspecified relation that groups those more specific. For example, if we ask the BKB what a “receiver” is, the most frequent relations are:

   ```
   290 'person':is:'receiver'
   29 'player':is:'receiver'
   16 'pick':is:'receiver'
   15 'one':is:'receiver'
   14 'receiver':is:'target'
   8 'end':is:'receiver'
   7 'back':is:'receiver'
   6 'position':is:'receiver'
   ```

   The number indicates the number of times the relation appears explicitly in the collection.

2. Has-instance: between an entity class and an entity instance. For example, if we ask for instances of team, the top 10 instances with more support in the collection are:

   ```
   192 'team':has-instance:'Jets'
   189 'team':has-instance:'Giants'
   43 'team':has-instance:'Eagles'
   40 'team':has-instance:'Bills'
   36 'team':has-instance:'Colts'
   35 'team':has-instance:'Miami'
   35 'team':has-instance:'Vikings'
   34 'team':has-instance:'Cowboys'
   32 'team':has-instance:'Patriots'
   31 'team':has-instance:'Dallas'
   ```

But we can ask also for the possible classes of an instance. For example, all the entity classes for “Eric_Davis” are:

```
12 'cornerback':has-instance:'Eric_Davis'
1 'hand':has-instance:'Eric_Davis'
1 'back':has-instance:'Eric_Davis'
```

There are other lexical relations as “part-of” and “is-value-of” in which we are still working. For example, the most frequent “is-value-of” relations are:

```
5178 '[0-9]-[0-9]':is-value-of:'lead'
3996 '[0-9]-[0-9]':is-value-of:'record'
2824 '[0-9]-[0-9]':is-value-of:'loss'
1225 '[0-9]-[0-9]':is-value-of:'season'
```

4 Enrichment procedure

The goal of the enrichment procedure is to determine what kind of events and entities are involved in the text, and what semantic relations are hidden by some syntactic dependencies such as noun-noun compound or some prepositions.

4.1 Fusion of nodes

Sometimes, the syntactic dependency ties two or more words that form a single concept. This is the case with multiword terms such as “tight end”, “field goal”, “running back”, etc. In these cases, the meaning of the compound is beyond the syntactic dependency. Thus, we shouldn’t look for its explicit meaning. Instead, we activate the fusion of the nodes into a single one.

However, there are some open issues related to the cases were fusion is not preferred. Otherwise, the process could be done with standard measures like mutual information, before the parsing step (and possibly improving its results).

The question is whether the fusion of the words into a single expression allows or not the consideration of possible paraphrases. For example, in the case of “`field:nn:goal`”, we don’t find other ways to express the concept in the BKB. However, in the case of “`touchdown:nn:pass`” we can find, for example, “`pass:for:touchdown`” a significant amount of times, and we want to identify them as equivalent expressions. For this reason, we find not convenient to fuse these cases.

4.2 Building context for instances

Suppose we wish to determine what kind of entity “`Steve Walsh`” is in the context of the syntactic dependency “`Steve_Walsh:nn:pass`”. First, we look into the BKB for the possible entity classes of `Steve_Walsh` previously found in the collection. In this particular case, the most frequent class is “`quarterback`”:

```
40 'quarterback':has-instance:'Steve_Walsh'
2 'junior':has-instance:'Steve_Walsh'
```

But, what happens if we see “`Steve_Walsh`” for the first time? Then we need to find evidence from other entities in the same syntactic context. We found that “`Marino`”, “`Kelly`”, “`Elway`”, “`Dan_Marino`”, etc. appear in the same kind of proposition (“`N:nn:pass`”) where we found “`Steve_Walsh`”, each of them supported by 24, 17, 15 and 10 occurrences respectively. However, some of the names can be ambiguous. For example, searching for “`Kelly`” in our BKB yields:

```
153 'quarterback':has-instance:'Jim_Kelly'
19 'linebacker':has-instance:'Joe_Kelly'
17 'quarterback':has-instance:'Kelly'
14 'quarterback':has-instance:'Kelly_Stouffer'
10 'quarterback':has-instance:'Kelly_Ryan'
8 'quarterback':has-instance:'Kelly_Holcomb'
7 'cornerback':has-instance:'Brian_Kelly'
```

Whereas others are not so ambiguous:

```
113 'quarterback':has-instance:'Dan_Marino'
6 'passer':has-instance:'Dan_Marino'
5 'player':has-instance:'Dan_Marino'
```

Taking this into account, we are able to infer that the most plausible class for an entity involved in a “`NNP:nn:pass`” proposition is a quarterback.

4.3 Building context for dependencies

Now we want to determine the meaning behind such syntactic dependencies as “`Steve_Walsh:nn:pass`”, “`touchdown:nn:pass`“, “`Young:nn:pass`” or “`pass:to:Brent_Jones`”. We have two ways for adding more meaning to these syntactic dependencies: find the most appropriate prepositions to describe them, and find the most appropriate verbs. Whether one, the other or both is more useful has to be determined during the reasoning system development.

Finding the prepositions

There are several types of propositions in the BKB that involve prepositions. The most relevant are NPN and NVNPN. In the case of “touchdown:nn:pass”, preposition “for” is clearly the best interpretation for the “nn” dependency:

```
NPN 712 'pass':'for':'touchdown'
NPN 24 'pass':'include':'touchdown'
NPN 3 'pass':'with':'touchdown'
NPN 2 'pass':'of':'touchdown'
NPN 1 'pass':'in':'touchdown'
NPN 1 'pass':'follow':'touchdown'
NPN 1 'pass':'to':'touchdown'
```

In the case of “`Steve_Walsh:nn:pass`” and “`Young:nn:pass`”, assuming they are quarterbacks, we can ask for all the prepositions between “pass” and “quarterback”:

```
NPN 23 'pass':'from':'quarterback'
NPN 14 'pass':'by':'quarterback'
NPN 2 'pass':'of':'quarterback'
NPN 1 'pass':'than':'quarterback'
NPN 1 'pass':'to':'quarterback'
```

Notice how lower frequencies involve more noisy options.

If we don’t have any evidence on the instance class, and we know only that they are instances, the pertinent query to the BKB obtains:

```
NPN 1305 'pass':'to':'NNP'
NPN 1085 'pass':'from':'NNP'
NPN 147 'pass':'by':'NNP'
NPN 144 'pass':'for':'NNP'
```

In the case of “`Young:nn:pass`” (in “Young pass to Brent Jones”), there exists already the preposition “to” (“pass:to:Brent_Jones”), so the most promising choice become the second, “pass:from:Young”, which has one order of magnitude more occurrences than the following.

In the case of “Steve_Walsh:nn:pass” (in “Eric Davis intercepted a Steve Walsh pass”) we can use additional information: we know that “`Eric_Davis:intercept:pass`”. So, we can try to find the appropriate preposition using NVNPN propositions in the following way:
`Eric_Davis:intercept:pass:P:Steve_Walsh”`

Asking the BKB about the propositions that involve two instances with “intercept” and “pass” we get:

```
NVNPN 48 'NNP':'intercept':'pass':'by':'NNP'
NVNPN 26 'NNP':'intercept':'pass':'at':'NNP'
NVNPN 12 'NNP':'intercept':'pass':'from':'NNP'
```

We could also query the BKB with the classes we already found for “Eric_Davis” (cornerback, player, person):

```
NVNPN 11 'person':'intercept':'pass':'by':'NNP'
NVNPN 4 'person':'intercept':'pass':'at':'NNP'
NVNPN 2 'person':'intercept':'pass':'in':'NNP'
NVNPN 2 'person':'intercept':'pass':'against':'NNP'
NVNPN 1 'cornerback':'intercept':'pass':'by':'NNP'
```

All these queries accumulate evidence over a correct preposition “by” (“pass:by:Steve_Walsh”). However, an explicit entity classification would make the procedure more robust.

Finding the verbs

Now the exercise is to find a verb able to give meaning to the syntactic dependencies such as “`Steve_Walsh:nn:pass`”, “`touchdown:nn:pass`“, “`Young:nn:pass`” or “`pass:to:Brent_Jones`”.

We can ask the BKB what instances (NNP) do with passes. The most frequent propositions are:

```
NVN 2241 'NNP':'catch':'pass'
NVN 2106 'NNP':'throw':'pass'
NVN 844 'NNP':'complete':'pass'
NVN 434 'NNP':'intercept':'pass'

NVNPN 758 'NNP':'throw':'pass':'to':'NNP'
NVNPN 562 'NNP':'catch':'pass':'for':'yard'
NVNPN 338 'NNP':'complete':'pass':'to':'NNP'
NVNPN 255 'NNP':'catch':'pass':'from':'NNP'
```

Considering the evidence of “Brent_Jones” being instance of “end” (tight end), if we ask the BKB about the most frequent relations between “end” and “pass” we find:

```
NVN 28 'end':'catch':'pass'
NVN 6 'end':'drop':'pass'
```

So, in this case, the BKB suggests that the syntactic dependency “`pass:to:Brent_Jones`” means “Brent_Jones is an end catching a pass”. Or in other words, that “Brent_Jones” has a role of “catch-ER” with respect to “pass”.

If we want to accumulate more evidence on this we can consider NVNPN propositions including touchdown. We only find evidence for the most general classes (NNP and person):

```
NVNPN 189 'NNP':'catch':'pass':'for':'touchdown'
NVNPN 26 'NNP':'complete':'pass':'for':'touchdown'

NVNPN 84 'person':'catch':'pass':'for':'touchdown'
NVNPN 18 'person':'complete':'pass':'for':'touchdown'
```

This means, that when we have “touchdown”, we don’t have counting for the second option “Brent_Jones:drop:pass”, while “catch” becomes stronger.

In the case of “Steve_Walsh:nn:pass” we hypothesize that “Steve_Walsh” is a quarterback. Asking the BKB about the most plausible relation between a quarterback and a pass we find:

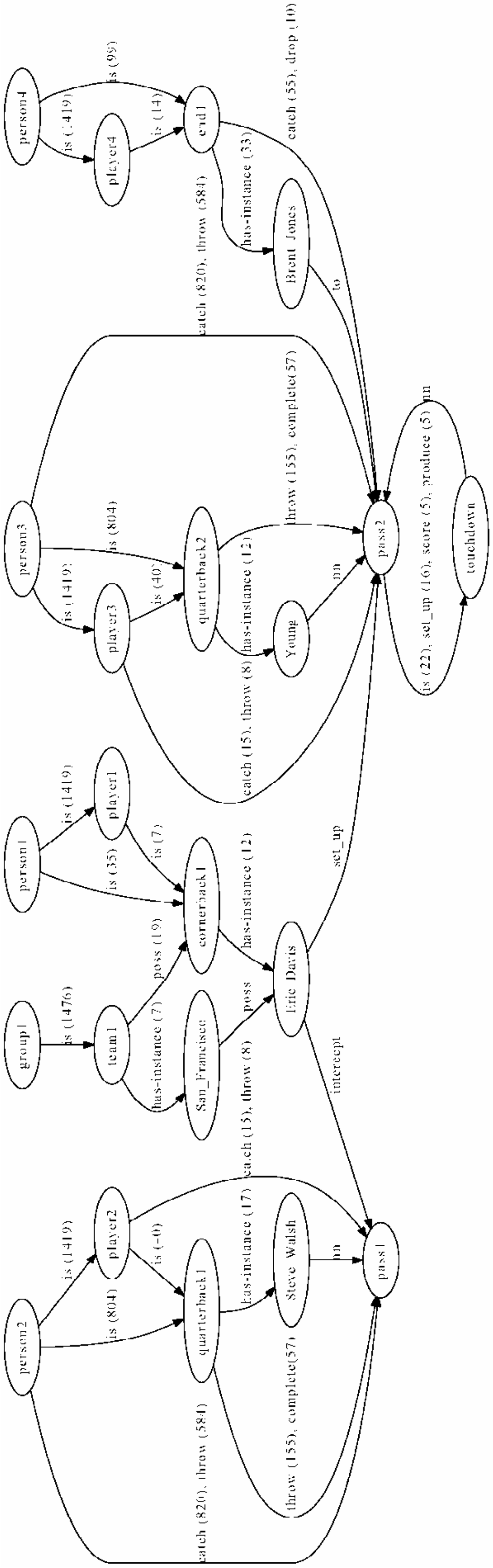

Figure 2. Graphical representation of the enriched text.

NVN 98 'quarterback':'throw':'pass'
NVN 27 'quarterback':'complete':'pass'

Again, if we take into account that it is a “`touchdown:nn:pass`”, then only the second option “`Steve_Walsh:complete:pass`” is consistent with the NVNPN propositions.

So, in this case, the BKB suggests that the syntactic dependency “`Steve_Walsh:nn:pass`” means “Steve_Walsh is a quarterback completing a pass”.

Finally, with respect to “`touchdown:nn:pass`“, we can ask about the verbs that relate them:

NVN 14 'pass':'set_up':'touchdown'
NVN 6 'pass':'score':'touchdown'
NVN 5 'pass':'produce':'touchdown'

Figure 2 shows the graphical representation of the sentence after some enrichment.

4.4 Expansion of relations

Sometimes, the sentence shows a verb with several arguments. In our example, we have “`Eric_David:intercept:pass:on:series`”. In these cases, the relation can be expanded and become a node.

In our example, the new node is the eventuality of “intercept” (let’s say “intercept-ION”), “Eric_Davis” is the “intercept-ER” and “pass” is the “intercept-ED”. Then, we can attach the missing information to the new node (see Figure 3).

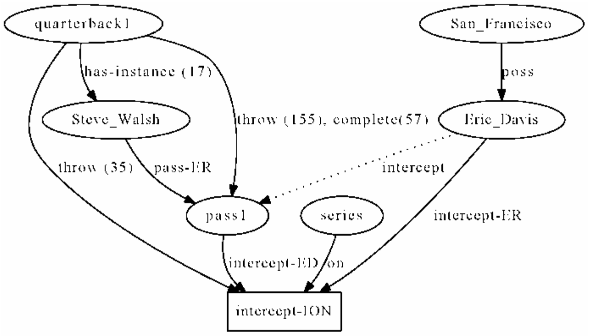

Figure 3. Expansion of the "intercept" relation.

In addition, we can proceed with the expansion of the context considering this new node. For example, we are working with the hypothesis that “Steve_Walsh” is an instance of quarterback and thus, its most plausible relations with pass are “throw” and “complete”. However, now we can ask about the most frequent relation between “quarterback” and “interception”. The most frequent is “`quarterback:throw:interception`” supported 35 times in the collection. From this, two actions can be done: reinforce the hypothesis of “`throw:pass`” instead of “`complete:pass`”, and add the hypothesis that
“`Steve_Walsh:throw:interception`”.

Finally, notice that since “set_up” doesn’t need to accommodate more arguments, we can maintain the collapsed edge.

4.5 Constraining the interpretations

Some of the inferences being performed are local in the sense that they involve only an entity and a relation. However, these local inferences must be coherent both with the sentence and the complete document.

To ensure this coherence we can use additional information as a way to constrain different hypotheses. In section 4.3 we showed the use of NVNPN propositions to constrain NVN ones.

Another example is the case of “`Eric_Davis:intercept:pass`”. We can ask the BKB for the entity classes that participate in such kind of proposition:

NVN 75 'person':'intercept':'pass'
NVN 14 'cornerback':'intercept':'pass'
NVN 11 'defense':'intercept':'pass'
NVN 8 'safety':'intercept':'pass'
NVN 7 'group':'intercept':'pass'
NVN 5 'linebacker':'intercept':'pass'

So the local inference for the kind of entity “Eric_Davis” is (cornerback) must be coherent with the fact that it intercepted a pass. In this case “cornerback” and “person” are properly reinforced. In some sense, we are using these additional constrains as shallow selectional preferences.

5 Evaluation

The evaluation of the enrichment process is a challenge by itself. Eventually, we will use extrinsic measures such as system performance on a QA task, applied first after reading a text, and then a second time after the enrichment process. This will measure the ability of the system to absorb and use knowledge across texts to enrich the interpretation of the target text. In the near term, however, it remains unclear which intrinsic evaluation measures to apply. It is not informative simply to count the number of additional relations one can attach to representation elements, or to count the increase in degree of interlinking of the nodes in the representation of a paragraph.

6 Related Work

To build the knowledge base we take an approach closely related to DART (Clark and Harrison, 2009) which in turn is related to KNEXT (Van Durme and Schubert, 2008). It is also more distantly related to TextRunner (Banko et al. 2007).

Like DART, we make use of a dependency parser instead of partial parsing. So we capture phrase heads instead complete phrases. The main differences between the generation of our BKB and the generation of DART are:

1. We use the dependencies involving copulative verbs as a source of evidence for "is" and "has-instance" relations.
2. Instead of replacing proper nouns by "person", "place", or "organization", we consider all of them just as instances in our BKB. Furthermore, when a proposition contains a proper noun, we count it twice: one as the original proposition instance, and a second replacing the proper nouns with a generic tag indicating that there was a name.
3. We make use of the modifiers that involve an instance (proper noun) to add counting to the "has-instance" relation.
4. Instead of replacing pronouns by "person" or "thing", we replace them by "person", "group" or "thing", taking advantage of the preposition number. This is particular useful for the domain of football where players and teams are central.
5. We add a new set of propositions that relate two clauses in the same sentence (e.g., `Floyd:break:takle:add:touchdown`). We tagged these propositions NVV, NVNV, NVVN and NVNVN.
6. Instead of an unrestricted domain collection, we consider documents closely related to the domain in which we want to interpret texts.

The consideration of a specific domain collection seems a very powerful option. Ambiguity is reduced inside a domain so the counting for propositions is more robust. Also frequency distribution of propositions is different from one domain into another. For example, the list of the most frequent NVN propositions in our BKB (see Section 3.1) is, by itself, an indication of the most salient and important events in the American football domain.

7 Conclusion and Future Work

The task of inferring omitted but necessary information is a significant part of automated text interpretation. In this paper we show that even simple kinds of information, gleaned relatively straightforwardly from a parsed corpus, can be quite useful. Though they are still lexical and not even starting to be semantic, propositions consisting of verbs as relations between nouns seem to provide a surprising amount of utility. It remains a research problem to determine what kinds and levels of knowledge are most useful in the long run.

In the paper, we discuss only the propositions that are grounded in instantial statements about players and events. But for true learning by reading, a system has to be able to recognize when the input expresses general rules, and to formulate such input as axioms or inferences. In addition, augmenting that is the significant challenge of generalizing certain kinds of instantial propositions to produce inferences. At which point, for example, should the system decide that "all football players have teams", and how should it do so? How to do so remains a topic for future work.

A further topic of investigation is the time at which expansion should occur. Doing so at question time, in the manner of traditional task-oriented back-chaining inference, is the obvious choice, but some limited amount of forward chaining at reading time seems appropriate too, especially if it can significantly assist with text processing tasks, in the manner of expectation-driven understanding.

Finally, as discussed above, the evaluation of our reading augmentation procedures remains to be developed.

Acknowledgments

We are grateful to Hans Chalupsky and David Farwell for their comments and input along this work. This work has been partially supported by the Spanish Government through the "Programa Nacional de Movilidad de Recursos Humanos del Plan Nacional de I+D+i 2008-2011 (Grant PR2009-0020). We acknowledge the support of DARPA contract number: FA8750-09-C-0172.

References

1. Banko, M., Cafarella, M., Soderland, S., Broadhead, M., Etzioni, O. 2007. Open Information Extraction from the Web. IJCAI 2007.
2. Barker, K. 2007. Building Models by Reading Texts. Invited talk at the AAAI 2007 Spring Symposium on Machine Reading, Stanford University.
3. Clark, P. and Harrison, P. 2009. Large-scale extraction and use of knowledge from text. The Fifth International Conference on Knowledge Capture (K-CAP 2009). http://www.cs.utexas.edu/users/pclark/dart/
4. Hobbs, J.R., Stickel, M., Appelt, D. and Martin, P., 1993. Interpretation as Abduction. Artificial Intelligence, Vol. 63, Nos. 1-2, pp. 69-142. http://www.isi.edu/~hobbs/interp-abduct-ai.pdf
5. Klein, D. and Manning, C.D. 2003. Accurate Unlexicalized Parsing. Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423-430
6. Marneffe, M. and Manning, C.D. 2008. The Stanford typed dependencies representation. In COLING 2008 Workshop on Cross-framework and Cross-domain Parser Evaluation.
7. Sperber, D. and Wilson, D. 1995. Relevance: Communication and cognition (2nd ed.) Oxford, Blackwell.
8. Van Durme, B., Schubert, L. 2008. Open Knowledge Extraction through Compositional Language Processing. Symposium on Semantics in Systems for Text Processing, STEP 2008.