Filling Knowledge Gaps in Text for Machine Reading

Anselmo Peñas
UNED NLP & IR Group
anselmo@lsi.uned.es

Eduard Hovy
USC Information Sciences Institute
hovy@isi.edu

Abstract

Texts are replete with gaps, information omitted since authors assume a certain amount of background knowledge. We define the process of enrichment that fills these gaps. We describe how enrichment can be performed using a Background Knowledge Base built from a large corpus. We evaluate the effectiveness of various openly available background knowledge bases and we identify the kind of information necessary for enrichment.

1 Introduction: Knowledge Gaps

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 pre-specified 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 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 of *enrichment*, which recognizes where gaps exist and fills them out using appropriate background knowledge as needed. We define *enrichment* as:

Def: Enrichment is the process of adding explicitly to a text's representation the information that is either implicit or missing in the text.

Central to enrichment is the source of the new knowledge. The use of Proposition Stores as Background Knowledge Bases (BKB) have been argued to be useful for improving parsing, coreference resolution, and word sense disambiguation (Clark and Harrison 2009). We argue here that Proposition Stores are also useful for Enrichment and show how in Section 4. However, we show in Section 5 that current BKB resources such as TextRunner (Banko et al. 2007) and DART (Clark and Harrison 2009) are not ideal for enrichment purposes. In some cases there is a lack in normalization. But the most important shortcoming is the lack in answering about instances, their possible classes, how they relate to propositions, and how different propositions are related through them. We propose easy to achieve extensions in this direction. We test this hypothesis building our own Proposition Store with the proposed extensions, and compare it with them for enrichment in the US football domain.

To perform enrichment, we begin with an initial simple text representation and a Proposition Stores as a background knowledge base. We execute a simple formalized procedure to select and attach appropriate elements from the BKB 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.

We describe in Section 2 our textual representations and in Section 3 the process of building the Proposition Store. Enrichment is described in Section 4, and an evaluation and comparison is performed in Section 5.

2 Text Representation

The initial, shallow, text representation must capture the first impression of what is going on in the text, possibly (unfortunately) losing some fragments. After the first impression, in accord with the purpose of the reading, we "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 it will become apparent whether we need to recover some of the missed elements, whether we need to expand some others, etc. So the process is identified with the growing of the context until deeper interpretation is possible. This approach resembles 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 (our Proposition Store) 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, being the object of the preposition the target of the relation.

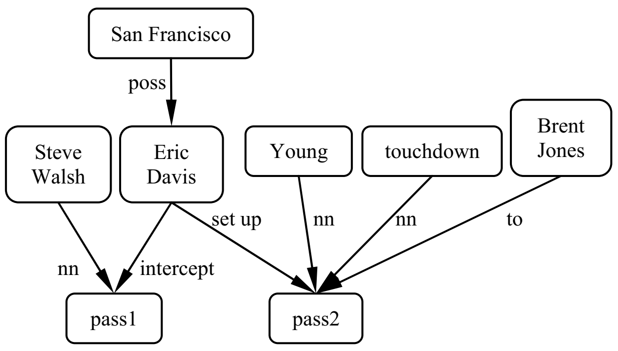

Figure 1. Initial text representation.

We 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 missing in the initial representation of the text, as for example "*on the next series*" or *"seven-yard"*.

3 Background Knowledge Base

We will use a Proposition Stores as a Background Knowledge Base (BKB). We built it from a collection of 30,826 New York Times articles about US football, similar to the kind of texts we want to interpret. We parsed the collection using a standard dependency parser (Marneffe and Manning, 2008; Klein and Maning, 2003) and, after collapsing some syntactic dependencies, obtained 3,022,305 raw elements in the BKB.

3.1 Types of elements in the BKB

We distinguish three kinds of elements in our Background Knowledge Base: Entities, Propositions, and Lexical relations. All three 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 nouns. We don't restrict classes to any particular predefined set. 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'

The 'NNP' tag replaces specific proper nouns (instances) 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 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 an entity instance. The enrichment of the text is based on the use of most frequent proposition classes.

Lexical Relations: We make use of very general patterns considering appositions and copula verbs (detected by the Stanford 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, it 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 frequency of the relation 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 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'

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'

We still work on other lexical relations such as "part-of" and "is-value-of". 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 operations

The goal of the following enrichment operations is to make explicit what kind of semantic relations and entity classes are involved in the text.

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 fuse the nodes into a single one.

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.

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 take into account the classes shared by other instances in the same syntactic context. The most frequent are "*Marino*", "*Kelly*", "*Elway*", etc. From them we are able to infer the most plausible class for the new entity. In our example, *quarterback*:

20 'quarterback':has-instance:'Marino'
6 'passer':has-instance:'Marino'
...
17 'quarterback':has-instance:'Kelly'
6 'passer':has-instance:'Kelly'
...
16 'quarterback':has-instance:'Elway'
9 'player':has-instance:'Elway'

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 useful has to be determined during the reasoning system development.

Finding the prepositions

Several types of propositions in the BKB involve prepositions. The most relevant are NPN and NVNPN. In the case of "*touchdown:nn:pass*", "*for*" is clearly the best interpretation:

NPN 712 'pass':'for':'touchdown'
NPN 24 'pass':'include':'touchdown'

In the case of "*Steve_Walsh:nn:pass*" and "*Young:nn:pass*", since we know they are quarterbacks, we can ask for all the prepositions between "*pass*" and "*quarterback*":

NPN 23 'pass':'from':'quarterback'
NPN 14 'pass':'by':'quarterback'

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'

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 becomes the second, "*pass:from:Young*", which has one order of magnitude more occurrences than its successor.

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 obtain:

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 have 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 the preposition "*by*" ("*pass:by:Steve_Walsh*").

Finding the verbs

The next exercise is to find a verb able to give meaning to 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 counts 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:

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 resulting enrichment after the process described.

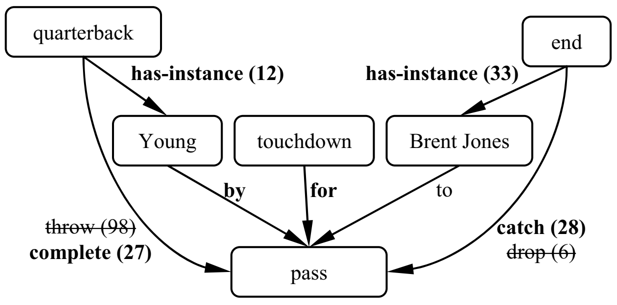

Figure 2. Enrichment of the noun phrase: "*Young touchdown pass to Brent Jones*"

4.4 Expansion of relations

Sometimes, the sentence shows a verb with more than two arguments. In our example, we have "*Eric_David:intercept:pass:on:series*". In these cases, relations can be expanded into new nodes.

Following our example, the new node is the eventuality of "*intercept*" ("*intercept-ION*"), "*Eric_Davis*" is the "*intercept-ER*" and "*pass*" is the "*intercept-ED*". Then, the missing information is attached to the new node (see Figure 3).

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 a nominalization of "intercept". The most frequent proposition is "*quarterback:throw:interception*", supported 35 times in the collection. In this way, we have inferred that the nominalization for the eventuality of intercept is interception (in our documents). Two further actions are possible: reinforce the hypothesis of "*throw:pass*" instead of "*complete:pass*" and add the hypothesis that "*Steve_Walsh:throw:interception*".

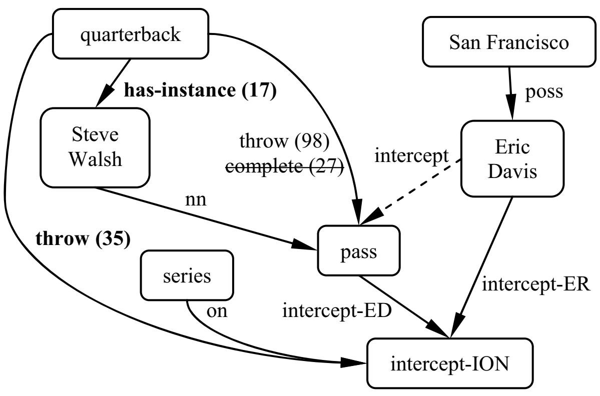

Figure 3. Expansion of "intercept" relation

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'

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 selectional preferences.

5 Evaluation

Properly evaluating the enrichment process is very difficult. Ideally, one would compare the output of an enrichment engine—a text graph fully fleshed out with additional knowledge—to a gold-standard graph containing all relevant information explicitly, and measure Recall and Precision of the links added by enrichment. But since we have no gold standard examples, and it is unclear how much knowledge should be included manually if one were to try to build some, two options remain: extrinsic evaluations and measuring the utility of the BKB in providing knowledge. We are in the process of performing an extrinsic evaluation, by measuring how much QA about the text read improves using the enriched representation. We report here the results of comparing the utility, for enrichment purposes, of two other publicly available background knowledge bases: DART (Clark and Harrison, 2009) and TextRunner (Banko et al. 2007).

5.1 Ability to answer about instances

As shown in our examples, BKBs need the ability to answer about instances and their classes. The BKBs don't need to be completely populated, but at least have enough instance-class attachments in order to allow analogy.

Neither DART nor TextRunner allow asking about possible classes for a particular instance. This is out of the scope of TextRunner. In DART, instances are replaced by one of three basic categories (person, place, organization). Although storing the original proper nouns attached to the assigned class would be straightforward, these three general classes are not enough to support inference. This leads us to the next ability.

5.2 Ability to discover new classes and relations

While *quarterbacks throw passes*, *ends* usually *catch or drop* them. As we have shown in our examples, classifying them as "*person"* or even "*player"* is not specific enough for enrichment.

Using a predefined set of entity classes doesn't seem a good approach for midterm goals. First, human abstraction is not correlated with the appropriate granularity level that enable recovering of relevant background knowledge. Second, annotation will be needed for training.

In our Proposition Stores, we count simply what is explicitly said in the texts about our instances. This seems correlated to an appropriate level of granularity. Furthermore, an instance can be attached to several classes that can be compatible (quarterback, player, person, leader, veteran, etc.). Frequencies tell us the classes we have to consider in the first place in order to find a coherent interpretation of the text.

5.3 Ability to constrain interpretation and accumulate evidence

Enrichment must be guided by the coherence of the ensuing interpretation. For this reason BKBs must allow different types of queries over the same elements. The aim is to constrain as much as possible the relations we recover to the ones that give a coherent interpretation of the text.

As shown in our example, we require the ability to ask different syntactic contexts/structures (NN, NVNPN, etc.), not only NVN (subject-verb-object). Achieving this is very difficult for approaches that don't use parsing.

5.4 Ability to digest enough knowledge adapted to the domain

None of the abilities discussed above are relevant if the BKB doesn't contain enough knowledge about the domain in which we want to enrich documents. To evaluate, we ran three simple queries related to the US football domain in order to assess the suitability of the BKBs for enrichment: *What do quarterbacks do with passes? What do persons do with passes? Who intercepts passes?* Table 1 shows the results obtained with DART, TextRunner and our BKB.

Although DART is a general domain BKB built using parsing, its approach doesn't allow one to process enough information to answer the first question (first row in Table 1). A web scale resource such as TextRunner is better suited for this purpose. However, results show its lack of normalization. On the other hand, our BKB is able to return a clean and relevant answer.

The second question (second row) shows the ability of the three BKBs to deal with a basic abstraction needed for inference. Since TextRunner doesn't perform any kind of processing over entities or pronouns, it doesn't recover relevant knowledge for this question in the football domain. In addition, the table shows the need for domain adaptation: most of the TextRunner relations, such as "person:gets:pass" or "person:bought:pass", refer to different domains. DART shows the same effect: the first two entries ("person:make:pass", "person:take:pass") belong to different domains.

DART[1]	TextRunner[2]	BKB (Football)
(no results)	(~200) threw (~100) completed (36) to throw (26) has thrown (19) makes (19) has (18) fires	(99) throw (25) complete (7) have (5) attempt (5) not-throw (4) toss (3) release
(47) make (45) take (36) complete (30) throw (25) let (23) catch (1) make (1) expect	(22) gets (17) makes (10) has (10) receives (7) who has (7) must have (6) acting on (6) to catch (6) who buys (5) bought (5) admits (5) gives	(824) catch (546) throw (256) complete (136) have (59) intercept (56) drop (39) not-catch (37) not-throw (36) snare (27) toss (23) pick off (20) run
(13) person (6) person/ place/ organization (2) full-back (1) place	(30) Early (26) Two plays (24) fumble (20) game (20) ball (17) Defensively	(75) person (14) cornerback (11) defense (8) safety (7) group (5) linebacker

Table 1. Comparison of DART, TextRunner and our BKB for the following queries (rows): (1) *quarterback:X:pass*, (2) *person:X:pass*, (3) *X:intercept:pass*. Frequencies are in parentheses.

Finally, the third question is aimed at recovering possible agents (those that *intercept passes* in our case). Again, as shown in DART, the reduced set of classes given by the entity recognizer is not enough for the football domain. But having no classes (TextRunner) is even worse, showing its orientation to discovering relations rather than to generalizing and answering about their possible arguments. Our approach is able to discover plausible agent-classes for the query.

Other queries related to the football domain show the same behavior.

[1] Available at http://userweb.cs.utexas.edu/users/pclark/dart/
[2] After aggregating partial results for each cluster using http://www.cs.washington.edu/research/textrunner/

6 Related Work

Our approach lies between macro-reading and Open Information Extraction (OIE). Macro-reading (Mitchell et al. 2009) is a different task from ours; it seeks to populate ontologies. Here concepts and relations are predefined by the ontology.

OIE (Banko et al. 2007) does not use a predefined set of semantic classes and relations and is aimed at web scale. For this reason the framework does not include a complete NLP pipeline. The resulting lack of term normalization and absence of domain adaptation (e.g., the query *person:X:pass* return *throw* but also *buy*) makes the results less relevant to single-document reading.

When, as with DART, the complete NLP pipeline is applied over a general corpus, the amount of information to be processed has to be limited due to computational cost. Ultimately, too little knowledge remains for working in a specific domain. For example, asking DART about "*quarterback:X:pass*" produces no results.

Our approach takes advantage of both worlds, ensuring that enough amounts of documents related to the domain will be processed with a complete NLP pipeline. Doing so provides cleaner and canonical representations (our propositions) and even higher counts than TextRunner for our domain. This level of processing will be scalable in the midterm; various people including (Huang and Sagae, 2010) are working in linear time parsers with state-of-the-art performance.

Another intermediate point between a collection of domain documents and the general web, reached by restricting processing to the results of a web query, is explored in IE-on-demand (Sekine 2006; Shinyama and Sekine 2006). However, they use a predefined set of entity classes, preventing from discovering the appropriate granularity level that enables retrieval of relevant background knowledge. We do not predefine the concepts/classes and relations, but discover them from what it is explicitly said in the collection.

The process of building the BKB described here is closely related to DART (Clark and Harrison, 2009) which in turn is related to KNEXT (Van Durme and Schubert, 2008). Perhaps the most important extension we performed is the inclusion of lexical relations (like "*has-instance*") that activate more powerful uses of the Proposition Stores.

7 Conclusions and Future Work

In building a BKB, limiting oneself to a specific domain provides some powerful benefits. Ambiguity is reduced inside the domain, making counts in propositions more accurate. Also, frequency distributions of propositions differ from one domain to 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 specifically in the US football domain. Furthermore, the amount of text required to build the BKB is reduced significantly allowing processing such as parsing.

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 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? This 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 intrinsic evaluation procedures remains to be developed.

Acknowledgments

We are grateful to Hans Chalupsky and David Farwell for their comments and input for this work. We acknowledge the builders of TextRunner and DART for their willingness to make their resources openly available.

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). Research supported in part by Air Force Contract FA8750-09-C-0172 under the DARPA Machine Reading Program.

References

Banko, M., Cafarella, M., Soderland, S., Broadhead, M., Etzioni, O. 2007. Open Information Extraction from the Web. IJCAI 2007.

Barker, K. 2007. Building Models by Reading Texts. Invited talk at the AAAI 2007 Spring Symposium on Machine Reading, Stanford University.

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/

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

Huang, L. and Sagae, K. 2010. Dynamic Programming for Linear-Time Shift-Reduce Parsing. ACL 2010.

Klein, D. and Manning, C.D. 2003. Accurate Unlexicalized Parsing. Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423-430

Marneffe, M. and Manning, C.D. 2008. The Stanford typed dependencies representation. In COLING 2008 Workshop on Cross-framework and Cross-domain Parser Evaluation.

Mitchell, T. M., Betteridge, J., Carlson, A., Hruschka, E., and Wang, R. Populating the Semantic Web by Macro-reading Internet Text. The Semantic Web - ISWC 2009. LNCS Volume 5823. Springer-Verlag.

Sekine, S. 2006. On Demand Information Extraction. COLING 2006.

Shinyama, Y. and Sekine, S. 2006. Preemptive Information Extraction using Unrestricted Relation Discovery. HLT-NAACL 2006.

Sperber, D. and Wilson, D. 1995. Relevance: Communication and cognition (2nd ed.) Oxford, Blackwell.

Van Durme, B., Schubert, L. 2008. Open Knowledge Extraction through Compositional Language Processing. Symposium on Semantics in Systems for Text Processing, STEP 2008.