Element level semantic matching using WordNet

Mikalai Yatskevich and Fausto Giunchiglia

Dept. of Information and Communication Technology,
University of Trento,
38050 Povo, Trento, Italy
{fausto,yatskevi}@dit.unitn.it

Abstract. We think of Match as an operator which takes two graph-like structures and produces a mapping between semantically related nodes. The matching process is essentially divided into two steps: element level and structure level. Element level matchers consider only labels of nodes, while structure level matchers start from this information to consider the full graph. In this paper we present and evaluate, on the large scale real world dataset, twelve new element level semantic matchers. The matchers exploit WordNet as a background knowledge source, and return semantic relations (e.g., equivalence, more general) between concepts rather than similarity coefficients between labels in the [0, 1] range. The twelve element level semantic matchers are evaluated against seven state of the art matching systems. The results of the matchers are found comparable with the results of the matching systems.

1 Introduction

We think of matching as the task of finding semantic correspondences between elements of two graph-like structures (e.g., conceptual hierarchies, database schemas or ontologies). Matching has been successfully applied to many well-known application domains, such as schema/ontology integration, data warehouses, and XML message mapping.

The matching task is often articulated into two basic steps, namely element and structure level matching (see [9, 26, 29] for a thorough discussion). Element level matchers consider only the information at the atomic level (e. g., the information contained in elements of the schemas), while structure level matchers consider also the information about the structural properties of the schemas.

Our goal in this paper is to describe a set of element level semantic matchers (i.e, the matchers that return a semantic relations (e.g., equivalence, more general, disjointness) rather then similarity coefficients [0..1]) and analyze their performance on the large scale real world dataset. The matchers exploit WordNet [21], a large repository of English lexical items, as a background knowledge source. We distinguish between two categories of matchers namely knowledge based and gloss based matchers. Knowledge based matchers produce the semantic relations exploiting the structural properties of WordNet often combined with statistics collected from large scale corpora. Some of the knowledge based

matchers are based on well known semantic similarity measures. The gloss based matchers exploit WordNet concept descriptions (or glosses). To the best of our knowledge gloss based matchers have never been applied to schema/ontology matching tasks, while some of knowledge based matchers, although being implemented within the matching systems, have never been evaluated on the commonly accepted datasets on its own.

The main contributions of the paper include:

- (i) a set of twelve new element level semantic matchers that exploit WordNet as a background knowledge source. These matchers encompass both new and well known semantic similarity measures and techniques.
- (ii) a thorough evaluation of the semantic element level matchers on a large scale real world dataset along with a comparative analysis of the evaluation results against the results of seven state of the art matching systems participated in the ontology matching evaluation OAEI-2006 [7].

The rest of the paper is organized as follows. Section 2 defines the basic notions while Section 3 and Section 4 are dedicated to knowledge and gloss based semantic element level matchers respectively. The evaluation results are discussed in Section 5. Section 6 provides an overview of the related work while Section 7 concludes the paper.

2 Semantic matching

2.1 Motivating scenario

In order to motivate the matching problem and illustrate one of the possible situations which can arise in the data integration task let us use the (parts of the Google and Yahoo) directories depicted in Figure 1. Suppose that the task is to

Fig. 1. Parts of Google and Yahoo directories

integrate these two directories. A first step in the integration process is to identify

the matching candidates. For example, $Shopping_{O1}$ can be assumed equivalent to $Shopping_{O2}$, while $Board\_Games_{O1}$ is less general than $Games_{O2}$. Hereafter the subscripts designate the directory (either O1 or O2) of the node considered. Once the correspondences between two schemas have been determined, the next step has to generate query expressions that automatically translate data instances of these schemas under an integrated schema.

We think of a *mapping element* (or *mapping*) as a 4-tuple $\langle ID_{ij}, n1_i, n2_j, R\rangle$, $i = 1, ..., N_1$; $j = 1, ..., N_2$; where ID_{ij} is a unique identifier of the given mapping element; $n1_i$ is the i-th node of the first graph, N_1 is the number of nodes in the first graph; $n2_j$ is the j-th node of the second graph, N_2 is the number of nodes in the second graph; and R specifies a similarity relation of the given nodes. For instance, in this paper we consider equivalence ($\equiv$); more general ($\sqsupseteq$); less general ($\sqsubseteq$); disjointness ($\perp$) relations. The semantics of the above relations are the obvious set-theoretic semantics. When none of the relations holds, the special *Idk* (I do not know) relation is returned. We define *matching* as the process of discovering mappings between two graph-like structures through the application of a matching algorithm.

2.2 Element level semantic matching

The matching process is often articulated into two basic steps, namely element and structure level matching (See [26, 29] for thorough discussion):

- *Element level matchers* consider only the information at the atomic level (e. g., the information contained in elements of the schemas);
- *Structure level matchers* often aggregate the results of the several element level matchers and consider also the information about the structural properties of the schemas.

Element level semantic matchers return semantic relations ($\equiv$,$\sqsubseteq$,$\sqsupseteq$,$\perp$,*Idk*) rather then similarity coefficients [0..1] which are often considered as equivalence relation with certain level of plausibility or confidence (see [9] for detailed discussion). In this paper we consider two classes of element level semantic matchers, namely:

- *Knowledge based matchers* take in input two concept (or synset) identifiers defined in WordNet. They produce semantic relations by exploiting its structural properties. In some cases they combine the knowledge derived from WordNet with statistics collected from large scale corpora. Often knowledge based matchers are based on either similarity or relatedness measures. If the value of the measure exceeds the given threshold the certain semantic relation is produced. Otherwise *Idk* is returned.
- *Gloss based matchers*, similarly to knowledge based matchers, take two concept (synset) identifiers as an input and return the semantic relation holding between them. However, gloss based matchers differ in that they use the information contained in natural language concept descriptions such as WordNet glosses.

Table 1 illustrates element semantic level matchers.

Table 1. Element level semantic matchers

Matcher name	Matcher type	Exploits
The WordNet matcher(WN)	Knowledge based	WordNet structure
The Leacock Chodorow matcher(LCM)		WordNet structure + Corpora statistics
The Resnik matcher(RM)		
The Jiang Conrath matcher(JCM)		
The Lin matcher(LM)		
The Hirst-St.Onge matcher(HOM)		
The Context Vectors matcher(CV)		
The WordNet Gloss matcher(WNG)	Gloss based	WordNet glosses
The WordNet Extended Gloss matcher(WNEG)		
The Gloss Comparison matcher(GC)		
The Extended Gloss Comparison matcher(EGC)		
The Semantic Gloss Comparison matcher(SGC)		

3 Knowledge based matchers

3.1 The WordNet matcher

WordNet [21] is a lexical database which is available online[1] and provides a large repository of English lexical items. WordNet contains synsets (or senses), structures containing sets of terms with synonymous meanings. Each synset has a gloss that defines the concept that it represents. For example, the words *night*, *nighttime* and *dark* constitute a single synset that has the following gloss: *the time after sunset and before sunrise while it is dark outside*. Synsets are connected to one another through explicit semantic relations. Some of these

Table 2. Relations in WordNet

Relation	Description	Example
Hypernym	is a generalization of	*motor vehicle* is a hypernym of *car*
Hyponym	is a kind of	*car* is a hyponym of *motor vehicle*
Meronym	is a part of	*lock* is a meronym of *door*
Holonym	contains part	*door* is a holonym of *lock*
Troponym	is a way to	*fly* is a troponym of *travel*
Antonym	opposite of	*stay in place* is an antonym of *travel*
Attribute	attribute of	*fast* is an attribute of *speed*
Entailment	entails	*calling on the phone* entails *dialing*
Cause	cause to	*to hurt* causes *to suffer*
Also See	related verb	*to lodge* is related to *reside*
Similar to	similar to	*evil* is similar to *bad*
Participle of	is participle of	*stored* is the participle of *to store*
Pertainym	pertains to	*radial* pertains to *radius*

[1] http://wordnet.princeton.edu/

relations (hypernymy, hyponymy for nouns and hypernymy and troponymy for verbs) constitute kind-of (or is-a) and part-of (holonymy and meronymy for nouns) hierarchies. For example, *tree* is a kind of *plant*, *tree* is hyponym of *plant* and *plant* is hypernym of *tree*. Analogously, from *trunk* is a part of *tree*, we have that *trunk* is meronym of *tree* and *tree* is holonym of *trunk*. The relations of WordNet 2.0 are presented on Table 2.

Figure 2 shows an example of nouns taxonomy.

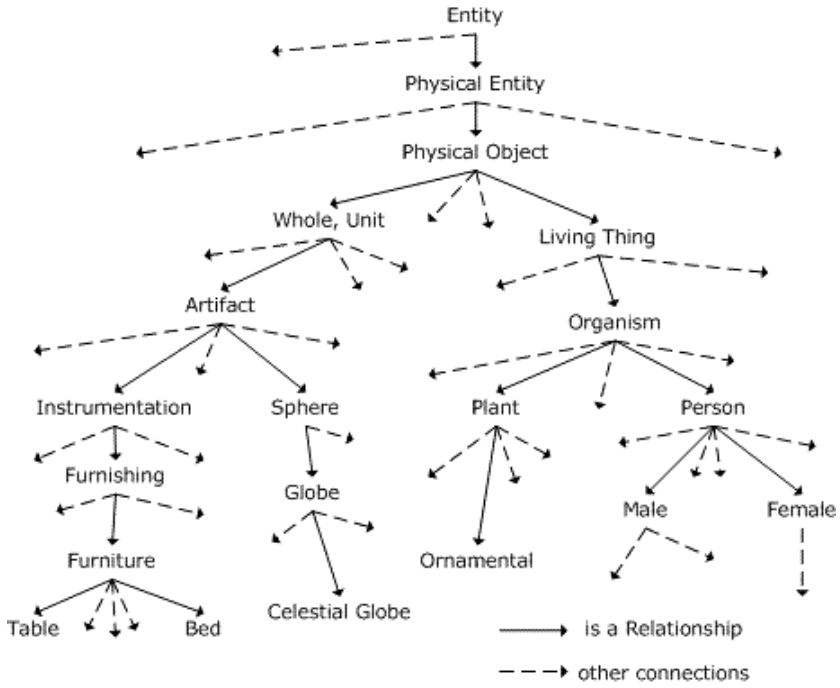

Fig. 2. An example of WordNet nouns taxonomy

The WordNet matcher is a knowledge based matcher. It translates the relations provided by WordNet to semantic relations according to the following rules:

- $A \sqsubseteq B$ if A is a hyponym, meronym or troponym of B;
- $A \sqsupseteq B$ if A is a hypernym or holonym of B;
- $A \equiv B$ if they are connected by synonymy relation or they belong to one synset (*night* and *nighttime* from abovementioned example);
- $A \perp B$ if they are connected by antonymy relation or they are the siblings in the part of hierarchy.

Notice that hyponymy, meronymy, troponymy, hypernymy and holonymy relations are transitive. Therefore, for example, from Figure 2 we can derive that $Person \sqsubseteq LivingThing$.

If none of the abovementioned relations holds among the two input synsets *Idk* relation is returned.

Table 3 illustrates WordNet matcher results.

Table 3. Semantic relations produced by the WordNet matcher

Source label	Target label	Semantic relation
car	minivan	$\sqsupseteq$
car	auto	$\equiv$
tail	dog	$\sqsubseteq$
red	pink	Idk

3.2 The Leacock Chodorow matcher

The Leacock Chodorow matcher exploits Leacock Chodorow semantic similarity measure [16]. It returns $\equiv$ if the measure exceeds the given threshold and *Idk* otherwise. The measure is based on counting the number of links between two input synsets. Intuitively, the shorter the path, the more related are the concepts under consideration. Leacock and Chodorow considered the noun *is a* hierarchy. They proposed the following formula for estimating the similarity of two synsets:

$$sim_{lc}(c_1, c_2) = -\ln\left(\frac{spath\,(c_1, c_2)}{2 \cdot D}\right) \tag{1}$$

where $spath(s_1, s_2)$ is the length of the shortest path between the two synsets c_1 and c_2 and D is the depth of the tree.

The measure has a lower bound of 0 and upper bound defined as follows

$$U_b = -\ln(1/(2 \cdot maxDepth)) \tag{2}$$

where $maxDepth$ is a maximum depth of the taxonomy.

Table 4 illustrates Leacock Chodorow matcher results with threshold 3.0.

Table 4. Semantic relations produced by the Leacock Chodorow matcher

Source label	Target label	Semantic relation
autograph	signature	$\equiv$
actor	actress	$\equiv$
dog	cat	Idk
sky	atmosphere	Idk

3.3 The Resnik matcher

The Resnik matcher exploits Resnik semantic similarity measure [27]. It returns $\equiv$ if the measure exceeds the given threshold and *Idk* otherwise. This measure is based on the concept of *information content* [27]. The information content defines the generality or specificity of a concept in a certain topic. The information content of the given concept is calculated as follows. Firstly the frequency[2] of concept occurrences F_C in the given *text corpus* is calculated. Then the frequencies of all subsuming concepts are calculated and added to F_C. Thus the root concept will count the occurrences of all the concepts in its taxonomy. In the case of WordNet synsets the frequency counts are precomputed for wide range of large scale corpora. We exploited *Brown corpus of standard american english* [15].

The information content of a concept c is defined as follows:

$$IC(c) = -\ln\left(\frac{freq(c)}{freq(root)}\right) \tag{3}$$

where $freq(c)$ and $freq(root)$ are, respectively, the frequencies of the concept c and the *root* of the taxonomy. Notice that the fraction represents the probability of occurrence of the concept in a large corpus.

Resnik defines the semantic similarity of the two concepts as the amount of information they share in common. To be more precise, the amount of information two concepts share in common is equal to the value of information content of their *lowest common subsumer*, that is the lowest node in the taxonomy that subsumes both concepts. For example, the lowest common subsumer of *cat* and *dog* is *carnivore*. Therefore, Resnik measure is defined as follows:

$$sim_{res}(c_1, c_2) = IC\left(lcs\left(c_1, c_2\right)\right) \tag{4}$$

where IC is the information content of a concept and $lcs(c_1, c_2)$ is the lowest common subsumer of concepts c_1 and c_2.

This measure has a lower bound of 0 and no upper bound.

Table 5 illustrates Resnik matcher results with threshold 10.0.

Table 5. Semantic relations produced by the Resnik matcher

Source label	Target label	Semantic relation
robot	android	$\equiv$
actor	actress	Idk
dog	cat	Idk

[2] Here and further in the paper, following the natural language understanding community tradition, we treat frequency as count (i.e., frequency of concept occurrences is a number of times the given concept occurs in the corpora).

3.4 The Jiang Conrath matcher

The Jiang Conrath matcher exploits Jiang Conrath semantic similarity measure [14]. It returns ≡ if the measure exceeds the given threshold and *Idk* otherwise. This measure incorporates both information content of the concepts and the information content of their lowest common subsumer. Originally Jiang Conrath defined the *distance* between two concepts as follows:

$$distance_{jc}(c_1, c_2) = IC(c_1) + IC(c_2) - 2 \cdot IC\left(lcs\left(c_1, c_2\right)\right) \tag{5}$$

where IC is the information content of a concept and lcs finds the lowest common subsumer of two given concepts.

Therefore the similarity of two concepts can be represented as

$$sim_{jc}(c_1, c_2) = \frac{1}{distance_{jc}(c_1, c_2)} \tag{6}$$

The formula has two special cases:

- In the first case all information content values are equal to 0:

$$IC(c_1) = IC(c_2) = IC\left(lcs\left(c_1, c_2\right)\right) = 0 \tag{7}$$

  This happens when both concepts and their lowest common subsumer are either the *root node* or have a frequency count of 0. In both cases 0 similarity is returned.

- The second case is when

$$IC(c_1) + IC(c_2) = 2 \cdot IC\left(lcs\left(c_1, c_2\right)\right) \tag{8}$$

  which usually happens when

$$IC(c_1) = IC(c_2) = IC\left(lcs\left(c_1, c_2\right)\right) \tag{9}$$

  In this case c_1 and c_2 are the same concept and so we would like to return a maximum value of relatedness.

This measure has a lower bound of 0 and the upper bound

$$U_b = \frac{1}{-\ln((f_{root} - 1)/f_{root})} \tag{10}$$

where f_{root} is the frequency of the taxonomy root.

The pseudo code for the algorithm is presented in Figure 3.

ID (line 2) is the unique identifier, **lemmas** (line 3) is the list of synonyms that represent this synset, **gloss** (line 4) is the definition associated to that synset and **relations** (line 5) is a list of pointers to other synsets connected to this by a WordNet relation. **maxValue** (line 12) is the upper bound.

Firstly the shortest path between two synsets is computed (line 7). Then the lowest common subsumer of the input synsets is obtained (line 8). The information content values are computed for both synsets and lowest common subsumer in lines 6-8. Finally after handling the special cases (lines 12-15) the **distance** (line 16) and similarity (line 17) are computed.

Table 6 illustrates Jiang Conrath matcher results with 1.0 threshold.

```
1 struct Synset
2   String ID;
3   String[] lemmas;
4   String gloss;
5   String[] relations;

6 float match( Synset synset1, Synset synset2 )
7   String shortestPathId = getShortestPath( synset1, synset2 );
8   Synset lcs = getLowCommonSubsumer( shortestPathId );
9   float ic_lcs = getInformationContent( lcs );
10  float ic_s1 = getInformationContent( synset1 );
11  float ic_s2 = getInformationContent( synset2 );
12  if ( ic_s1 == ic_s2 && ic_s1 == ic_lcs && ic_lcs == 0)
13   return 0;
14  if ( ic_s1 + ic_s2 == 2*ic_lcs )
15    return maxValue;
16  float distance = ic_s1 + ic_s2 - 2*ic_lcs;
17  return 1/distance;
```

Fig. 3. The Jiang Conrath matcher pseudo code

Table 6. Semantic relations produced by the Jiang Conrath matcher

Source label	Target label	Semantic relation
trip	hallucination	$\equiv$
actor	actress	$\equiv$
dog	cat	Idk

3.5 The Lin matcher

The Lin matcher exploits Lin semantic similarity measure [18]. It returns $\equiv$ if the measure exceeds the given threshold and *Idk* otherwise. This measure is also based on information content. It is defined as follows:

$$sim_{lin}(c_1, c_2) = \frac{2 \cdot IC\left(lcs\left(c_1, c_2\right)\right)}{IC(c_1) + IC(c_2)} \tag{11}$$

In the case of $IC(c_1) = 0$ and $IC(c_2) = 0$, 0 similarity is returned.

Table 7 illustrates the Lin matcher results with 0.9 threshold.

3.6 The Hirst-St.Onge matcher

The Hirst-St.Onge matcher exploits Hirst-St.Onge semantic similarity measure [12]. It returns $\equiv$ if the measure exceeds the given threshold and *Idk* otherwise. This measure in contrast to information content based measures is not restricted to WordNet noun hierarchies.

Hirst and St.Onge classified links in WordNet in 3 different categories:

Table 7. Semantic relations produced by the Lin matcher

Source label	Target label	Semantic relation
robot	android	Idk
actor	actress	≡
dog	cat	Idk

- *upward* (e.g. hypernym);
- *downward* (e.g. holonym);
- *horizontal* (e.g. antonym);

According to them two words are connected by a *strong* relation if:

- they both belong to the same synset;
- they belong to synsets connected by a *horizontal* link;
- one is a compound word, the second one is substring of the first while their synsets are connected by an *is a* relation;

For strongly related words relatedness is computed as $2 \cdot C$, where C is a constant used in the formula for medium-strong relations. Its default value is 8, so the coefficient is equal to 16.

A *medium-strong* relation exists if the two synsets are connected by a *valid path* in WordNet. A path is considered valid if it is not longer than 5 links and conforms to one of the eight predefined patterns. The relatedness between two words connected by a medium-strong relation corresponds to the *weight* of the path which is given by the following formula:

$$Weight = C - PathLength - k \cdot ChangesInDirection \tag{12}$$

where C and k are constants and, in our case they are assumed to be equal 8 and 1 respectively. The pseudo code below illustrates the algorithm. In order to compute the relatedness of two concepts first the type of relation holding between them is determined (line 2). Then the given relation is compared with existing patterns (line 4).

```
1 float match( Synset synset1, Synset synset2 )
2   if ( checkStrongRelationship( synset1, synset2 ))
3     return 2*C;
4   int weight = getMedStrongWeight( 0, 0, 0, synset1, synset2 );
5   return weight;
```

Fig. 4. The Hirst-St.Onge matcher pseudo code

checkStrongRelationship (line 2) takes in input the two synsets and returns true if they fulfill one of the requirements of *Strong relations* and false otherwise.

getMedStrongWeight (line 4) takes in input two synsets and three zero values, that are respectively defined as **state**, **distance** and **chdir**, and returns the weight as stated in Eq. 12 . Basically it searches recursively for a path from **synset1** to **synset2**, taking trace of the **distance**, changes in direction (**chdir**) so far and giving the **state** value for the next call. There are 8 possible states (from 0 to 7) in which the function may find itself, every state defines the rules the path has to follow. In particular, they specify the categories of links used in the last iteration, the ones that are allowed in the current step and the ones to use as next, taking into account the possible changes in direction.

This measure has a lower bound of 0 and an upper bound of 16.

Table 8 illustrates Hirst-St.Onge matcher results with threshold 4.0.

Table 8. Semantic relations produced by the Hirst-St.Onge matcher

Source label	Target label	Semantic relation
school	private school	≡
actor	actress	≡
dog	cat	Idk
sky	atmosphere	Idk

3.7 The Context Vectors matcher

The Context Vectors matcher exploits a context vectors semantic similarity measure [25]. It returns ≡ if the measure exceeds the given threshold and *Idk* otherwise. This measure is based on context vector notion introduced by Schütze in [28]. Originally exploited for *word sense disambiguation* context vectors was adapted for semantic similarity computation exploiting WordNet in [25].

The context vectors computation process starts from selection of the highly topical words which will define the dimensions of our *word space*. For our experiments we used WordNet glosses as a corpus. We stemmed all the words in the glosses and filtered out the function words. Afterwards we counted the frequencies of the words in the corpus. Then we cut off the words with frequencies lower than 5 and higher than 1000. This allowed us to keep the most informative words. We also added a *tf-idf* cutoff with an upper bound of 1500. tf-idf is a weight used to evaluate how important a word is to a document (or gloss in our case). The formula we used is as follows

$$tfidf = tf \cdot \ln(idf) \tag{13}$$

where tf is the frequency of occurence of the word and idf is defined as follows

$$idf = \frac{nr.documents}{docFrequency} \tag{14}$$

We used *nr.documents* as the number of glosses and *docFrequency* is the number of glosses in which our word appears. tf-idf cut off threshold allowed to perform

additional filtering of the frequent words. Further we will call the remaining words *content words*.

Afterwards we have created word vectors for all content words w as follows:

1. Initialize a vector $\overrightarrow{w}$ to zero;
2. Find every occurrence of w in WordNet glosses;
3. For each occurrence, search that gloss for words in the word space and increment the dimensions of $\overrightarrow{w}$ that correspond to those words.

The basic idea here is to have a matrix of word vectors, where every row corresponds to a word in the content words list and every column corresponds to the respective frequencies of each word in the word space.

The final step is to calculate gloss vectors for every synset in WordNet, this is done by adding the word vectors for each content word in the gloss. For example, for the gloss vector of *clock* we have to consider its gloss: *a timepiece that shows the time of day* and add the word vectors of *timepiece*, *shows*, *time* and *day*. Notice that this is a simplified example because for our experiments we use *extended glosses*, thus we had to take into account also the glosses of every concept connected to *clock* by a WordNet relation.

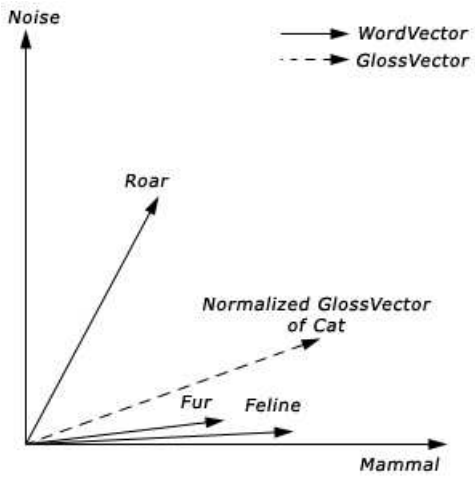

Fig. 5. An example of a 2-dimensional space for Word and Gloss vectors for "cat - feline mammal usually having thick soft fur and being unable to roar" synset

As soon as gloss vectors are calculated they are stored in a database[3]. Semantic similarity of two synsets is defined as follows

$$sim_{cv}(c_1, c_2) = \cos(angle(\overrightarrow{v_1}, \overrightarrow{v_2})) \tag{15}$$

[3] Notice that gloss vectors can be calculated once and further reused by semantic similarity computation algorithm.

where c_1 and c_2 are the concepts, $\overrightarrow{v_1}$ and $\overrightarrow{v_2}$ are the respective gloss vectors and *angle* is the angle between vectors. Eq. 15 can be rewritten using vector products, it becomes:

$$sim_{cv}(c_1, c_2) = \frac{\overrightarrow{v_1} \cdot \overrightarrow{v_2}}{|\overrightarrow{v_1}|\,|\overrightarrow{v_2}|} \tag{16}$$

where at the denominator we have the *magnitude*[4] of the two vectors. Eq. 16 is a dot product[5] between two normalized vectors.

Figure 5 illustrates context vectors similarity in a 2 dimensional space.

The pseudo code of context vector semantic similarity computation algorithm is as follows:

```
1 float match( Synset synset1, Synset synset2 )
2   int glossVec1[] = loadGlossVector( synset1 );
3   int glossVec2[] = loadGlossVector( synset2 );
4   float normVec1[] = normalizeVec( glossVec1 );
5   float normVec1[] = normalizeVec( glossVec1 );
6   return dotProduct( normVec1, normVec2 );
```

Fig. 6. Context vector semantic similarity computation pseudo code

loadGlossVector (line 2-3) loads, from the database, the precomputed gloss vector for the given synset. **normalizeVec** (line 4-5) calculates the normalized form of the given vector. **dotProduct** (line 6) return the dot product between two given vectors.

The measure has a lower bound of 0 and an upper bound of 1.

Table 9 illustrates context vectors matcher results with threshold 0.3.

Table 9. Semantic relations produced by the context vectors matcher

Source label	Target label	Semantic relation
autograph	signature	≡
actor	actress	≡
robot	android	≡
fruit	glass	Idk

[4] The magnitude of vector $\overrightarrow{v}$ is equal to $\sqrt{\sum_{i=1}^{n} v_i^2}$.
[5] The dot product between vector $\overrightarrow{v}$ and vector $\overrightarrow{w}$ is $\overrightarrow{v} \cdot \overrightarrow{w} = \sum_{i=1}^{n} v_i w_i$.

4 Gloss based matchers

4.1 The WordNet gloss matcher

The WordNet gloss matcher compares the labels of the first input sense with the WordNet gloss of the second. First, it extracts the labels of the first input sense from WordNet. Then, it computes the number of their occurrences in the second gloss. If this number exceeds a given threshold, $\sqsubseteq$ is returned. Otherwise, *Idk* is produced.

The reason why the less general relation is returned comes from the lexical structure of the WordNet gloss. Very often the meaning of the index words is explained through a specification of the more general concept. In the following example, *hound* (*any of several breeds of dog used for hunting typically having large drooping ears*) *hound* is described through the specification of the more general concept *dog*. In this example *hound* is a *dog* with special properties (*large drooping ears*, *used for hunting*).

Counting the label occurrences in the gloss does not give a strong evidence of what relation holds between concepts. For example, WordNet gloss returns the less general relation for *hound* and *ear* in the abovementioned example, which is clearly wrong.

Table 10 illustrates WordNet gloss matcher results.

Table 10. Semantic relations produced by the WordNet gloss matcher

Source label	Target label	Semantic relation
hound	dog	$\sqsubseteq$
hound	ear	$\sqsubseteq$
dog	car	Idk

4.2 The WordNet extended gloss matcher

The WordNet extended gloss matcher compares the labels of the first input sense with the extended gloss of the second. This extended gloss is obtained from the input sense descendants (ancestors) descriptions in the is-a (part-of) WordNet hierarchy. A given threshold determines the maximum allowed distance between these descriptions and the input sense in the WordNet hierarchy. By default, only direct descendants (ancestors) are considered. The idea of using extended gloss originates from [2]. Unlike [2], we do not calculate the extended gloss overlaps measure, but count the number of first input sense labels occurrences in the extended gloss of the second input sense. If this number exceeds a given threshold, a semantic relation is produced. Otherwise, *Idk* is returned. The type of relation produced depends on the glosses we use to build the extended gloss. If the extended gloss is built from descendant (ancestor) glosses, then the $\sqsupseteq$ ($\sqsubseteq$) relation is produced. For example, the relation holding between the words *dog*

and *breed* can be easily found by this matcher. These concepts are not related in WordNet, but the word *breed* occurs very often in the *dog* descendant glosses.

Table 11 illustrates WordNet extended gloss matcher results.

Table 11. Semantic relations produced by the WordNet extended gloss matcher

Source label	Target label	Semantic relation
dog	breed	$\sqsupseteq$
wheel	mashinery	$\sqsubseteq$
dog	cat	Idk

4.3 The Gloss comparison matcher

Within the Gloss comparison matcher the number of the same words occurring in the two input glosses increases the similarity value. The equivalence relation is returned if the resulting similarity value exceeds a given threshold. *Idk* is produced otherwise.

Let us try to find the relation holding, for example, between *Afghan hound* and *Maltese dog* using gloss comparison strategy. These two concepts are breeds of *dog*, but unfortunately WordNet does not have explicit relation between them. However, the glosses of both concepts are very similar. Let us compare:

Maltese dog is a breed of toy dogs having a long straight silky white coat.

And:

Afghan hound is a tall graceful breed of hound with a long silky coat; native to the Near East.

There are 4 shared words in both glosses (*breed*, *long*, *silky*, *coat*). Hence, the two concepts are taken to be equivalent. Table 12 illustrates gloss comparison matcher results. Several modifications of this matcher exist. One can assign a

Table 12. Semantic relations produced by the gloss comparison matcher

Source label	Target label	Semantic relation
Afghan hound	Maltese dog	$\equiv$
dog	cat	Idk

higher weight to the phrases or particular parts of speech than single words [24]. In the current implementation we have exploited the approach used in [24], but changed the output to be a semantic relation.

4.4 The Extended Gloss comparison matcher

The extended gloss comparison matcher compares two extended glosses built from the input senses. Thus, if the first gloss has a lot of words in common with

descendant glosses of the second then the first sense is more general than the second and vice versa. If the extended glosses formed from descendant (ancestor) glosses of both labels have a lot of words in common (this value is controlled by a given threshold) then the equivalence relation is returned. For example, *dog* and *cat* are not connected by any relation in WordNet. Comparing the extended glosses obtained from descendants glosses of both concepts we can find a lot of words in common (*breed*, *coat*, etc). Thus, we can infer that *dog* and *cat* are related (they are both pets), and return the equivalence relation. The relations produced by the matcher are summarized in Table 13.

Table 13. Semantic relations produced by the extended gloss comparison matcher

Source label	Target label	Semantic relation
house	animal	Idk
dog	cat	$\equiv$

4.5 The Semantic Gloss comparison matcher

The Semantic Gloss comparison matcher maintains statistics not only for the same words in the input senses glosses (like in Gloss comparison) but also for words which are connected through is-a (part-of) relationships in WordNet. This can help finding the gloss relevance not only at the syntactic but also at the semantic level. In Semantic Gloss Comparison we consider synonyms, less general and more general concepts what (hopefully) lead to better results.

In the first step the glosses of both senses are obtained. Then, they are compared by checking which relations hold in WordNet between the words of both glosses. If there is a sufficient amount (in the current implementation this value is controlled by a threshold) of synonyms the equivalence relation is returned. In the case of a large amount of more (less) general words, the output is $\sqsupseteq$ ($\sqsubseteq$) correspondingly. *Idk* is returned if we have a nearly equal amount of more and less general words in the glosses or there are no relations between words in glosses. Table 14 contains the results produced by semantic gloss comparison matcher.

Table 14. Semantic relations produced by extended gloss comparison matcher

Source label	Target label	Semantic relation
dog	breed	$\sqsupseteq$
dog	cat	Idk
wheel	machinery	$\sqsubseteq$

5 Evaluation

5.1 The Dataset

We have exploited for evaluation the data set constructed from Google, Yahoo and Looksmart web directories as described in [1, 34]. This dataset consists of a set of graph like structures and a set of mappings that holds among the nodes of the structures often referred as a reference mapping set). The key idea of the data set construction methodology was to significantly reduce the search space for human annotators. Instead of considering the full mapping task which is very big (Google and Yahoo directories have up to $3*10^5$ nodes each: this means that the human annotators need to consider up to $(3 * 10^5)^2 = 9 * 10^{10}$ mappings), it uses semi automatic pruning techniques in order to significantly reduce the search space. For example, for the dataset described in [1] human annotators consider only 2265 mappings instead of the full mapping problem.

The reference mapping set of the dataset contains not only positive but also negative mappings. It allows to exploit the dataset for evaluation of both recall and precision. The key difference of the reference mapping [34] in respect to conventional ones (see [32] for example) is that it does not contain the complete set of reference mappings. Instead, the reference mapping is composed from two parts [34]:

- Representative subset of complete reference mapping. It contains the positive mappings (i.e., the mappings that hold for the matching task).
- Representative subset of negative mappings (i.e., the mappings that does not hold for the matching task).

The reference mapping is composed from 2265 positive and 2374 negative mappings. The dataset was used in OAEI-2005,2006 [8, 7] ontology matching evaluations. Therefore the results of element level matchers can be easily compared with results shown by the matching systems participated in OAEI evaluations.

5.2 The evaluation methodology

Most of the matchers described in Sections 3 and 4 produce a semantic relation by comparing an internal similarity measure with the given threshold. In fact only the WordNet and the WordNet Gloss matchers do not depend on the threshold values since their results does not depend on the internal similarity measures. Taking into account the considerations above we have decided to calculate matching quality measures for various values of the thresholds.

We have chosen Precision, Recall and F-Measure as matching quality measures as the commonly agreed ones. Precision varies in the [0,1] range; the higher the value, the smaller the set of wrong mappings (false positives) which have been computed. Precision is a correctness measure. Recall varies in the [0,1] range; the higher the value, the smaller the set of correct mappings (true positives) which have not found. Recall is a completeness measure. F-measure varies in the [0,1] range. The version computed here is the harmonic mean of precision and recall.

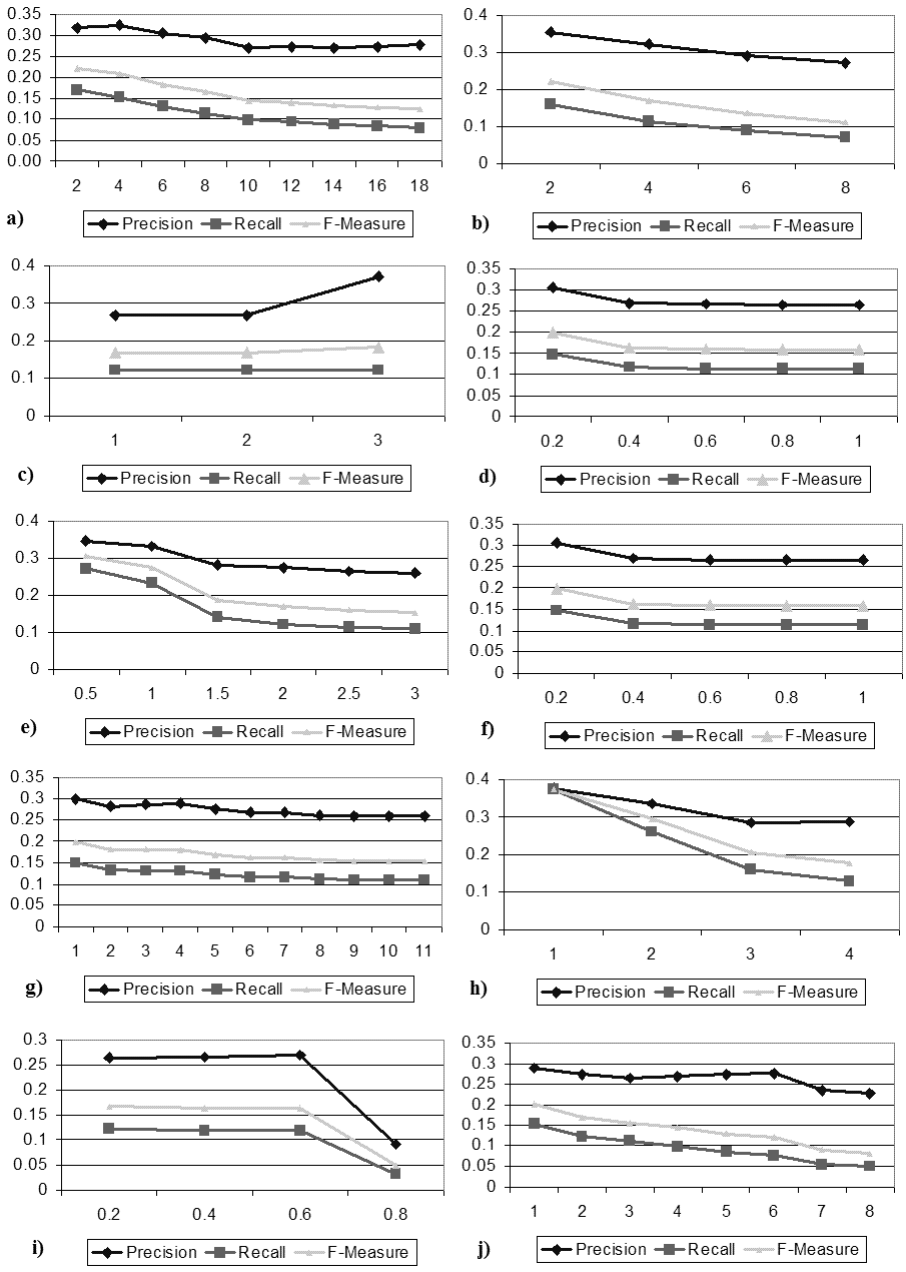

Fig. 7. Precision, Recall and F-Measure depending on threshold values for the : (a) Extended Gloss Comparison; (b) WordNet Extended Gloss; (c) Hirst-St.Onge; (d) Jiang Conrath; (e) Leacock Chodorow; (f) Lin; (g) Resnik; (h) WordNet Extended Gloss; (i) Context Vector; (j) Semantic Gloss Comparison matchers

It is global measure of the matching quality, growing with it. This choice opened to us the possibility to compare the results of element level semantic matchers with the results of the systems participated in OAEI ontology matching evaluations. We have chosen the best Precision, Recall and F-Measure for the matchers depending on their threshold values. We also have compared the results of the matchers with the results of the systems participated in OAEI-2006 ontology matching evaluation.

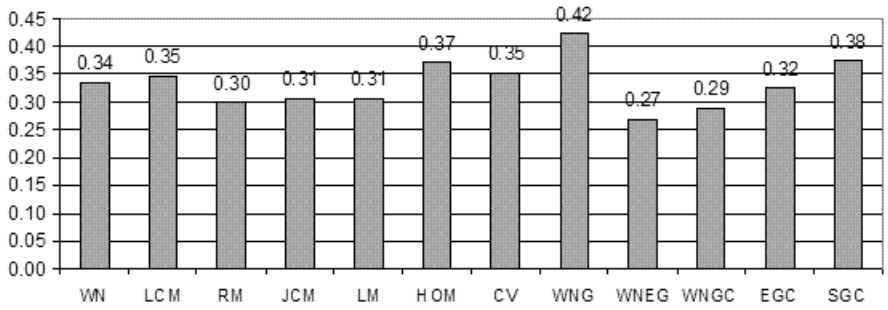

Fig. 8. Precision of element level semantic matchers

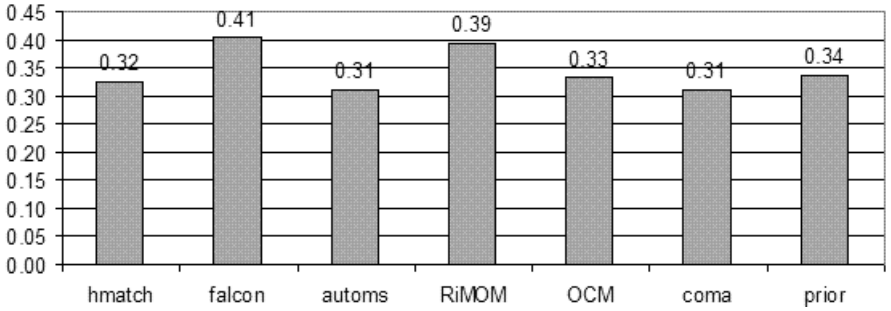

Fig. 9. Precision of matching systems participated in OAEI-2006 evaluation

Notice that both knowledge and gloss based matchers take two WordNet synset identifiers as an input. At the same time labels at nodes are expressed in natural language. Here and further in the paper we assume that all WordNet synsets are retrieved for the node labels. An element level semantic matcher finds a semantic relation holding between two given nodes if the matcher returns the relation for at least one of WordNet synsets pairs in the cross product of the synsets attached to the given nodes. Word sense disambiguation and filtering techniques (see [20] for example), in principle, may improve the results of the

matchers (see [10] for detailed discussion). However we have not used them in this evaluation.

5.3 Discussion

Figure 7 presents the results of the matchers depending on various threshold values. As form the figure the matchers demonstrate various behaviors depending on the threshold values. This fact is not surprising taking into account the different meaning of the threshold values for the different matchers. For example, for information content based matchers (such as Resnik, Jiang Conrath and Lin matchers) threshold value corresponds to the certain internal similarity measure values. At the same time the WordNet Gloss Comparison matcher considers the threshold as the number of the same words in the glosses of the synsets. The other observation is that similar matchers demonstrate the similar dependence on threshold values and even similar results. For example, Jiang Conrath and Lin matchers demonstrate almost the same results since they are both based on information content and exploit the similar ways of aggregating the information content values. In general the matchers demonstrate relatively high values of matching quality measures.

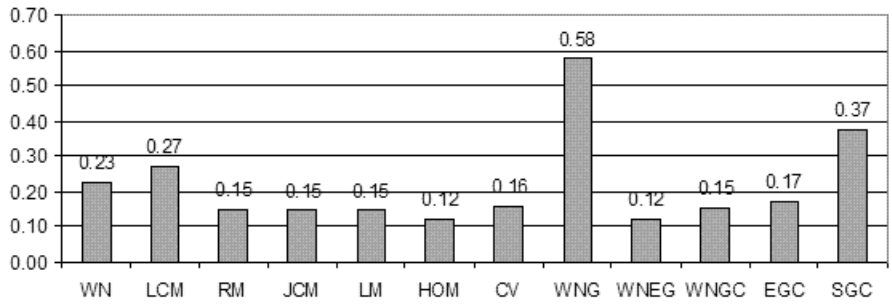

Fig. 10. Recall of element level semantic matchers

Figure 8 demonstrate the highest Precision values for all the matchers. Notice that matchers which do not depend on threshold values demonstrate a very high Precision results. For example, the WordNet Gloss matcher demonstrate the highest Precision among all the matchers while WordNet demonstrated sixth result. The second and third results were demonstrated by sophisticated semantic gloss comparison and Hirst - St.Onge matchers. Notice that the Precision values demonstrated by element level semantic matchers are comparable with the results of the matching systems on OAEI-2006 ontology matching evaluation. As from Figure 9, the best of the matching systems in OAEI-2006 evaluation demonstrated lower Precision than the best out of element level semantic matchers. Notice that strictly speaking the results of the matching systems are

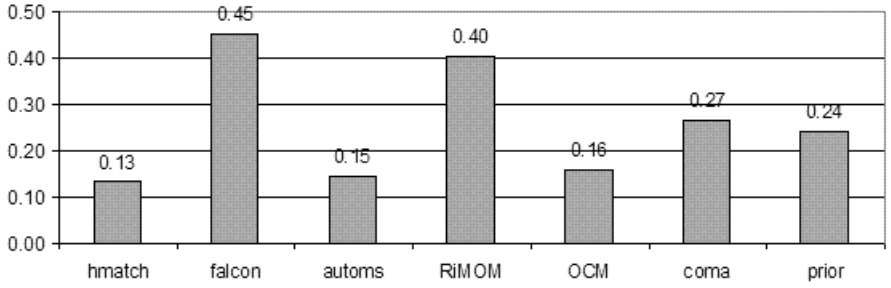

Fig. 11. Recall of matching systems participated in OAEI-2006 evaluation

not directly comparable with the results of threshold based matchers. The systems were evaluated blindly on the dataset [7]. Therefore the authors could not optimize the threshold values in the same way as we did for threshold based element level matchers (see Figure 7 for example). However the results illustrate the importance of WordNet in element level matching. They also emphasize the importance of the right aggregation strategy choice within the structure based matchers. Notice also that the matching systems were outperformed by threshold independent element level matcher (WordNet Gloss) whose results can not be optimized by tuning threshold values and therefore they are comparable with the systems results.

The Recall of the element level matchers is presented on Figure 10. Simi-

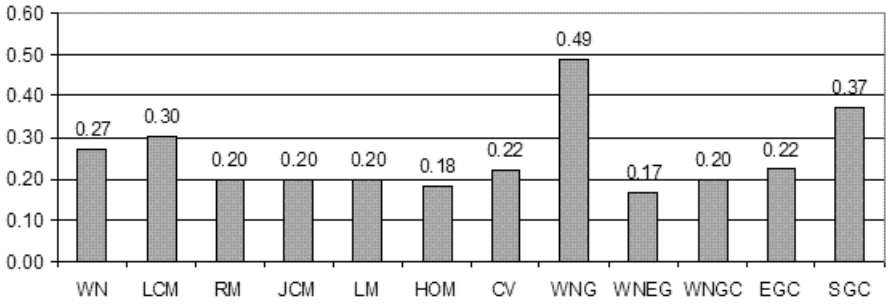

Fig. 12. F-Measure of element level semantic matchers

larly to Precision case the matchers that do not depend on the threshold values demonstrated the highest results. Recall of the WordNet Gloss matcher was the best while the WordNet matcher result is the fourth. Semantic gloss comparison and Hirst - St.Onge matchers demonstrated the second and the third results. Comparison with matching systems results is presented on Figure 11. As from the figure most of the element level semantic matchers produce the comparable to the matching systems results.

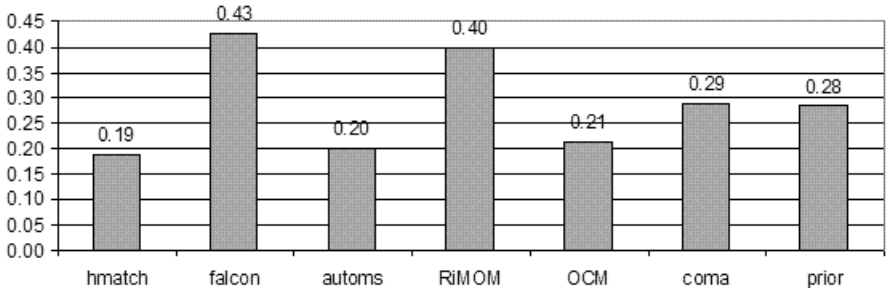

Fig. 13. F-Measure of matching systems participated in OAEI-2006 evaluation

A comparison between Figures 8, 10 and Figure 7 shows that the results of element level semantic matchers are comparable with the results of matching systems for the wide intervals of threshold values.

The F-Measure values for element level matchers and matching systems are presented on Figures 12 and 13 respectively. Similarly to Precision and Recall cases the best element level semantic matcher (WordNet Gloss) outperformed all the matching systems while all the other demonstrated the comparable performance.

6 Related Work

Development of element level matchers has focused on the variety of works originating from different communities. The term element level matching was coined in [26] while [9] introduced the notions of semantic matching and element level semantic matching. Early work in schema/ontology matching community was devoted to string based matchers or element level matchers exploiting various string similarity measures. For example, element level matchers based on Levenstain string edit distance [17] and nGrams have been used in [19, 11]. In [5] a library of string matching algorithms have been proposed. Among the others it contained a string edit distance algorithms proposed by Levenstain, Jaro [13], Winkler [33], Smith-Waterman [30] and string similarity algorithms of Jensen-Shannon [6], Monge-Elkan [22]. The new string distance metric and element level matcher implementing it have been proposed in [31]. The authors also evaluated the matcher on the subset of the matching tasks exploited in OAEI-2005 [8] evaluation and compared the results with Levenstain, Jaro-Winkler, Modge-Elkan, nGram, Smith-Waterman and Needleman-Wunsch [23] matchers. Differently from these matchers the element level semantic matchers described in the paper return semantic relations ($\equiv$,$\sqsubseteq$,$\sqsupseteq$,$\perp$,Idk) instead of numerical similarity coefficients and exploit WordNet as a background knowledge source.

While string based matchers have received a considerable attention in schema and ontology matching community knowledge and gloss based matchers have been largely overlooked until the very recent time. The WordNet matcher has

been introduced in [3] and further used in [10]. However the effectiveness of the matcher was largely unknown. It was not evaluated on its own, out of the systems comprising also various structure level algorithms. The library of element and structure level matchers has been recently presented in [35]. The library contains, among the others, Resnik and Lin matchers. However no qualitative measures of their effectiveness were reported. Differently from these works we focus on comparative evaluation of the matchers. The evaluation allows us to compare usefulness of the various techniques in real world matching scenarios and provides the useful hints to the system designers.

The measures of semantic similarity and relatedness such as Leacock Chodorow [16], Resnik [27], Jiang-Conrath [14], Lin [18], Hirst-St. Onge [12], Context Vectors [28, 25], extended gloss [2] have been introduced in the NLP community. They have been successfully applied to problems of word sense disambiguation, determining the structure of texts, text summarization and annotation, information extraction and retrieval, automatic indexing, lexical selection, and the automatic correction of word errors in text (see [4] for in depth discussion). Differently from these works we are focused on application of the semantic similarity and relatedness techniques to schema/ontology matching problems.

7 Conclusions

We have presented twelve new element level semantic matchers. The matchers exploit WordNet as a background knowledge source, and return semantic relations ($\equiv$,$\sqsubseteq$,$\sqsupseteq$,$\perp$,*Idk*) between concepts, rather than similarity coefficients between labels in the [0..1] range. We have evaluated the matchers on large scale real world dataset extracted from Google, Yahoo and Looksmart web directories. We have also compared the evaluation results with the results demonstrated by the matching systems in the ontology matching evaluation OAEI-2006. The results of the element level matchers are found comparable with the results of the matching systems. The results of this evaluation opened a number of questions regarding the effectiveness of state of the art structure level matching algorithms. In fact the WordNet Gloss element level matcher outperformed sophisticated matching systems comprising a set of both element and structure level matchers.

Future work includes analysis of the state of the art structure level matchers on various datasets. The results of this work will help to the development of robust matching algorithms. It also may provide useful insights to development of iterative and interactive matching systems. They will improve the quality of the mappings by iterating and by focusing user's attention on the critical points where his/her input is maximally useful. These advancements are hardly possible without a comprehensive testing methodology which is able to estimate quality of the mappings between schemas with hundreds and thousands of nodes. Initial steps have already been done; see for details [1]. Here, the key issue is that in these cases, specifying expert mappings manually is (often) neither desirable nor feasible task.

Acknowledgments

We would like to thank Pavel Shvaiko for insightful discussions and his work on S-Match.

This work has been partially supported by the European Knowledge Web network of excellence (http://knowledgeweb.semanticweb.org/) and the STReP OpenKnowledge (http://www.openk.org/).

References

1. P. Avesani, F. Giunchiglia, and M. Yatskevich. A large scale taxonomy mapping evaluation. In *Proceedings of International Semantic Web Conference (ISWC)*, pages 67–81, 2005.
2. S. Banerjee and T. Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 267–270, 2003.
3. P. Bouquet, L. Serafini, and S. Zanobini. Semantic coordination: a new approach and an application. In *The Semantic Web*, volume 2870 of *LNCS*, Sanibel Island, Fla., 20-23 October 2003.
4. A. Budanitsky and G. Hirst. Evaluating wordnet-based measures of semantic distance. *Computational Linguistics*, 32(1):13–47, 2006.
5. W. Cohen, P. Ravikumar, and S. Fienberg. A comparison of string metrics for matching names and records. In *Proceedings of the KDD 2003.*
6. I. Dagan, L. Lee, and F. C. N. Pereira. Similarity-based models of word cooccurrence probabilities. *Machine Learning*, 34(1-3):43–69, 1999.
7. J. Euzenat, Malgorzata Mochol, Ondrej Svab, Vojtech Svatek, Pavel Shvaiko, H. Stuckenschmidt, Willem Robert van Hage, and Mikalai Yatskevich. Introduction to the ontology alignment evaluation 2006. In *Proceedings of Ontology Matching 2006 Workshop at ISWC'06*, 2006.
8. J. Euzenat, H. Stuckenschmidt, and M. Yatskevich. Introduction to the ontology alignment evaluation 2005. In *Proceedings of K-CAP 2005 Workshop on Integrating Ontologies*, 2005.
9. F. Giunchiglia and P. Shvaiko. Semantic matching. *The Knowledge Engineering Review Journal*, (18(3)):265–280, 2003.
10. F. Giunchiglia, P. Shvaiko, and M. Yatskevich. S-Match: an algorithm and an implementation of semantic matching. In *Proceedings of 1st european semantic web symposium (ESWS'04)*, volume 3053 of *LNCS*, Heraklion, 10-12 May 2004.
11. H.H.Do and E. Rahm. COMA - a system for flexible combination of schema matching approaches. In *Proceedings of Very Large Data Bases Conference (VLDB)*, pages 610–621, 2002.
12. G. Hirst and D. St-Onge. Lexical chains as representation of context for the detection and correction malapropisms. In *C. Fellbaum, editor, WordNet: An electronic lexical database and some of its applications. Cambrige, MA: The MIT Press.*, 1997.
13. M. A. Jaro. Probabilistic linkage of large public health data files. *Statistics in Medicine*, 14(5–7):491–498, 1995.
14. J.J. Jiang and D.W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics*, 1998.

15. H. Kucera and W. N. Francis. *Computational analysis of present-day American English*. Providence, Brown University Press, 1967. http://CEUR-WS.org/Vol-128/.
16. C. Leacock, M. Chodorow, and G. A. Miller. Using corpus statistics and wordnet relations for sense identification. *Computational Linguistics*, 24(1):147–165, 1998.
17. V.I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Sov. Phys. Dokl.*, 6, 1966.
18. D. Lin. An information-theoretic definition of similarity. In *ICML*, pages 296–304, 1998.
19. J. Madhavan, P. A. Bernstein, and E. Rahm. Generic schema matching with cupid. pages 49–58, 2001.
20. B. Magnini, M. Speranza, and C. Girardi. A semantic-based approach to interoperability of classification hierarchies: Evaluation of linguistic techniques. In *Proceedings of COLING-2004*, August 23 - 27, 2004.
21. G. A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.
22. A. E. Monge and C. Elkan. The field matching problem: Algorithms and applications. In *Knowledge Discovery and Data Mining*, pages 267–270, 1996.
23. S.B. Needleman and C.S. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:443–453, 1970.
24. S. Patwardhan, S. Banerjee, and T. Pedersen. Using measures of semantic relatedness for word sense disambiguation. 2003.
25. S. Patwardhan and T. Pedersen. Using wordnet based context vectors to estimate the semantic relatedness of concepts. In *Proceedings of the EACL 2006 Workshop Making Sense of Sense - Bringing Computational Linguistics and Psycholinguistics Together*, pages 1–8, 2006.
26. E. Rahm and P. Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal*, (10(4)):334–350, 2001.
27. P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *IJCAI*, pages 448–453, 1995.
28. H. Schütze. Dimensions of meaning. In *SC*, pages 787–796, 1992.
29. P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. *Journal on Data Semantics*, IV:146–171, 2005.
30. T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.
31. G. Stoilos, G. Stamou, and S. Kollias. A string metric for ontology alignment. In *Proceedings of International Semantic Web Conference (ISWC)*, 2005.
32. Y. Sure, O. Corcho, J. Euzenat, and T. Hughes. *Evaluation of Ontology-based Tools*. Proceedings of the 3rd International Workshop on Evaluation of Ontology-based Tools (EON), 2004. http://CEUR-WS.org/Vol-128/.
33. W. Winkler. The state of record linkage and current research problems. In *Technical report, Statistical Research Division, U.S. Bureau of the Census, Wachington, DC*, 1999.
34. M. Yatskevich, F. Giunchiglia, and P. Avesani. A large scale dataset for the evaluation of matching systems. In *Technical Report. University of Trento*, 2005.
35. P. Ziegler, C. Kiefer, C. Sturm, K. Dittrich, and A. Bernstein. Detecting similarities in ontologies with the soqa-simpack toolkit. In *10th Int. Conference on Extending Database Technology (EDBT 2006).*, 2006.