Clustering of terms from translation dictionaries and synonyms lists to automatically build more structured Linguistic Resources

Maria Teresa Pazienza, Armando Stellato

AI Research Group, Dept. of Computer Science,
Systems and Production University of Rome, Tor Vergata
Via del Politecnico 1, 00133 Rome, Italy
{pazienza,stellato}@info.uniroma2.it

Abstract

Building a Linguistic Resource (LR) is a task requiring a huge quantitative of means, human resources and funds. Though finalization of the development phase and assessment of the produced resource, necessarily require human involvement, a computer aided process for building the resource's initial structure would greatly reduce the overall effort to be undertaken. We present here a novel approach for automatizing the process of building structured (possibly multilingual) LRs, starting from already available LRs and exploiting simple vocabularies of synonyms and/or translations for different languages. A simple algorithm for clustering terms, according to their shared senses, is presented in two versions, both for separating flat list of synonyms and flat lists of translations. The algorithm is then motivated against two possible exploitations: reducing the cost for producing new LRs, and linguistically enriching the content of existing semantic resources, like SW ontologies and knowledge bases. Empirical results are provided for two experimental setups: automatic term clustering for English synonyms list, and for Italian translations of English terms.

1. Introduction

Building a Linguistic Resource (LR) is a task requiring a huge quantitative of means, human resources and funds. The product of such enterprises is typically supposed to be a rich, complex and widely assessed resource which well deserves the cost it has been spent on its development.

WordNet (Fellbaum, 1998) and its multilingual cousins EuroWordnet (Vossen, 1998) and Balkanet (Stamou et al.), are probably good embodiments of this assertion, being the result of a huge combined effort of tens of researches from different academic institutions and research centers. The original WordNet, in particular, has seen the complexity of its schema and the size of its content continuously grow in the years of its existence, laying the basis for the following wordnet-like linguistic databases.

Though human involvement required by the task of building and maintaining such complex resources is hard to substitute, still something could be done to aid linguists in their work. A Computer Aided LR Development (CALD?) could be based on reuse of terms from simple existing resources (bilingual dictionaries, monolingual synonyms list, thesauruses etc…), to produce suggestions for developing more complex, wordnet-like, resources: building new synsets, adding new synonyms to existing synsets, and performing other operations which are needed during development/maintenance of a LR. These suggestions could then be prompted to the language engineers in a interactive process, and be accepted, modified or rejected upon their knowledge and good sense.

In this work we focus on one specific task, which could play a fundamental role in a CALD process: clustering linguistic expressions, retrieved from lists of synonyms/translations for a given term, upon their different senses. The aim of this task is to provide a more structured view over resources which bear interesting linguistic information, but expose their content in a scarcely organized manner. This structured view can then be exploited in a CALD process for developing, maintaining or extending new resources. The next two sections provide details about the task and propose dedicated algorithmic solutions, followed by section 4 which comments results of experimentation conducted on two popular language resources. Section 5 discusses possible applications of the presented techniques and proposed a further algorithm for developing new wordnets in different languages, while section 6 ends this work.

2. Motivations: A plethora of existing reusable resources

"The term linguistic resources refers to (usually large) sets of language data and descriptions in machine readable form, to be used in building, improving, or evaluating natural language (NL) and speech algorithms or systems" (Cole et al., 1997). This definition includes, among the others, lexical databases, bilingual dictionaries and terminologies.

Our motivations arised from the consideration that the Web is full of available linguistic resources, like DICT dictionaries (www.dict.org/bin/Dict), Freelang dictionaries (www.freelang.com), Babylon (www.babylon.com) compatible dictionaries etc…, ranging from simple, general purpose, bilingual dictionaries, to domain specific thesauruses.

These resources largely differentiate upon the explicit linguistic information they expose, which may vary in format, content granularity and motivation (linguistic theories, intended users, purpose or system-oriented scope etc…); in general, apart from a few noble (and rarely free) examples, they are often missing of any structural organization of their content, and are often unreliable, being the product of freely independent initiatives and of frequent updates by untitled volunteers.

Nonetheless, their content may represent an interesting (raw) source of information that, once filtered and restructured, could be exploited as a basis for developing/maintaining more complex LRs.

This work is focused on structuring the information of a class of LRs, which we named *flat* LRs, consisting in simple entries of the form:

`IndexWord: WORDLIST`

where IndexWord is a word (or complex linguistic expression) belonging to the "domain" of the resource, and WORDLIST is a flat collection of words/linguistic expressions pertaining to the given IndexWord. The semantics (i.e., what "pertaining" means) of this class of entries depends strictly on the nature of the resource; for instance, in bilingual dictionaries WORDLIST will contain translations of IndexWord, while in a dictionary of synonyms, WORDLIST will expose synonymic expressions for IndexWord, expressed in its same idiom.
In particular, we gave these resources the name of *flat LRs*, because the expressions contained in WORDLIST are not synonymic among them, or better, sometimes they do, sometimes they do not (depending on ambiguity of the IndexWord and of its polysemic expressions). For instance, if we consider the thesaurus of Microsoft Word, and we ask for synonyms of the word: *plane*, we get these results:

`plane: { airplane, hydroplane, aircraft, jet, seaplane, flat surface, level surface, flat }`

here it is clear that different, heterogeneous relationships exists between the terms in WORDLIST (or, better to say, the linguistic concepts which are supposed to represent their meaning in the given context). *Airplane*, *hydroplane*, *aircraft*, *jet* and *seaplane* are all terms identifying more specific concepts of the one expressed by *plane*, which (in the context of these terms) we could identify as "a vehicle that has a fixed wing and is powered by propellers or jets"; on a deeper investigation of these more specific terms, we could define *hydroplane* and *seaplane* as true synonyms, while *jet* denotes a more specific concept of *airplane*, which is in turn more specific both than *aircraft* and *plane*.
Seen as a wordnet structure, this part of the WORDLIST would appear this way:

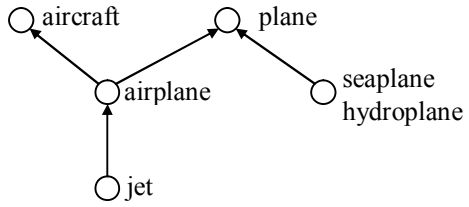

Figure 1: a synsets&synonyms representation for some of MS Word synonyms for term: plane

Anyway, on a first analysis, we could consider all the previous terms as related to a general conceptual cluster, which is quite distant from the meaning expressed by the remaining linguistic expressions. The two terms *flat surface* and *level surface* in fact, may be considered synonyms wrt a second sense of the word *plane*, while *flat* is the singleton of a third sense of this IndexWord, where it is used as an adjective to mean "something having a surface without slope"

3. Term clustering for flat LRs

In the previous example, the WORDLIST appeared as a flat list of linguistic expressions, without any separation for identifying terms pertaining to similar meanings. A thesaurus like the one used for the example, could prove to be useful in providing lexical entries to be put inside a more complex LR, and surely an application which both permits to browse its content, and modify it to create a new LR, would be appreciated by a (team of) linguist in doing its work. Language engineers would simply access the provided word knowledge and use their world knowledge to separate, structure and reorganize the browsed terms. However, we believe that a further step towards automatization of this task is still possible.

3.1. Term clustering, monolingual resources

We thus developed an algorithm for clustering terms in WORDLIST, upon their different meanings. The task is performed without knowledge provided by any external information sources. The concept behind the algorithm, which is based on finding overlaps on different sets of synonyms, is quite intuitive: to make an example, when we are looking for translations of a given term on a bilingual dictionary, and we want to be sure that our chosen translation is a proper one (i.e. a translation which reflects the sense of the term we are using in our context), we immediately search a reverse translation for it (in our language, which we are more confident with), and check if it is still consistent with the intended meaning.
In our case, we simply substitute the knowledge that a human has about its native language, with a cross checking of the result of reiterated queries, starting from the original term and carried over all of its synonyms/translations. Formally, in the case of a dictionary of synonyms, we have the following algorithm:

clusterSynonyms(*t:term*) SET OF *sense*
Begin
let senses(t) ← ∅
∀ t_i ∈ **R**.getPotSyns(t), unmarked(t_j)
  let sense ts ← { t_i }
  mark(t_i);
  // lookahead
  ∀ t_j ∈ **R**.getPotSyns(t), unmarked(t_j) |
    ∃ t_{jk} : t_{jk} ∈{**R**.getPotSyns(t_j)\ t_j}, t_{jk} ∈ terms(ts)
    do { **let** ts ← ts ∪{ t_j }; mark(t_j) }
  // lookback
  If { ∃ s ∈ senses(t), t_s ∈ terms(s),
      t_{sk} ∈ **R**.getPotSyns(t_s) |
      t_{sk} ∈ terms(ts) }
  Then let s ← s ∪ ts
  Else senses(t) ← senses(t) ∪ { ts }
  EndIf
Return senses(t)

Fig. 2: algorithm for term clustering, monolingual resources

where:

- **R**: is the Linguistic Resource which is being queried
- t: is the initial IndexTerm
- {…}: denotes a set of elements (they are reported inside the brackets)
- senses(term) SET OF *sense*: a sense is a cluster of terms which convey similar meaning. When called, this function reports all the clusters which have been *currently* identified.
- terms(s) SET OF *term*: when called, this function

term	t1s	t2s
imposta	sportello	imposta girevole
		anta
		battente
		porta
		portiere
		portello
		~~imposta~~
	battente	martellante
		continuo
		ininterrotto
		assillante
		dirotto
		scrosciante
		sferzante
		~~imposta~~
	persiana	~~imposta~~
		scuretto
		scura
	scuro	buio
		privo di luce
		povero di luce
		oscuro
		ombroso
		appannato
		opaco
		offuscato
	scuretto	~~imposta~~
		scuro
	anta	tavola
		sportello
		~~imposta~~
		porta
		battente
	tassa	imposizione
		contribuzione
		contributo
		trattenuta
		carica
		aggravio
		esazione
		tributo
	tributo	tassa
		~~imposta~~
		imposizione
		gravame
		gabella
		balzello
		contributo
		Contribuzione

Table 1: MS Word synonyms for the Italian word: "Imposta"

returns all the terms which have been currently collected in sense *s*.

- getPotSynonyms(term *t*) SET OF term: is a method of the resource, which takes an IndexWord as input and returns a WORDLIST.
- unmarked(term *t*) BOOLEAN: returns *false* if mark(term *t*) has previously been invoked on *t*
- mark(term *t*): is a function which puts a mark on *t*

We report here a real example, coming from MS Word thesaurus, where synonyms of the Italian word "Imposta" have been clustered wrt their different meanings. Column "t1s" of Table 1 reports the synonyms suggested for "Imposta", while column "t2s" contains a list of suggested synonyms, for each of the terms in "t1s". Here follows a step-by-step application of the algorithm to the given example (to save space, trivial passages are omitted):

senses(*Imposta*) = {}

examining: *sportello*:
lookahead: *sportello* ∈ **R**.getPotSynonyms(*anta*) **then**:
ts ← {*sportello, anta*}; mark(*anta*); senses(*Imposta*) ← { *ts* }

examining: *battente*:
lookback: *battente* ∈ **R**.getPotSynonyms(*sportello*) **then**:
{*sportello, anta*} ← {*sportello, anta*} ∪ { *battente* }

examining: *persiana*:
no matches neither in lookahead nor in lookback
senses(*imposta*) = {{*sportello*, *anta*, *battente*},{*persiana*}}

examining: *scuro*:
lookahead: *scuro* ∈ **R**.getPotSynonyms(*scuretto*) **then**:
ts ← {*scuro, scuretto*}; mark(*scuretto*);
lookback: *scuretto* ∈ R.getPotSynonyms(*persiana*) **then**:
{*persiana*} ← {*persiana*} ∪ { *scuro, scuretto* }
senses(*imposta*) = { {*sportello*, *anta*, *battente*},
{*persiana, scuro, scuretto* }}

examining: *tassa*:
lookahead: *tassa* ∈ **R**.getPotSynonyms(*tributo*) **then**:
ts ← {*tassa, tributo*}; mark(*tributo*);
senses(*imposta*) = { {*sportello*, *anta*, *battente*},
{*persiana, scuro, scuretto* },
{*tassa*. *tributo*} }

Figure 3: step-by-step execution of term clustering algorithm

As we can observe from the resulting set of terms, synonyms for "Imposta" have been correctly separated towards their different senses, by analyzing overlaps between them and the set of "second generation" synonyms. There is a homomorphism between this problem and a trivial specialization of the graph-partitioning problem. If we define:

- terms in <t1s> as vertices of the graph
- terms in <t2s> as edges of the graph
- and we state that an edge/*t2* connects two vertices/*t1s* if *t2* is identical to one of the *t1s* and is a synonym for the other

clusterTranslations(*t:term*) SET OF *sense*
Begin
let senses(*t*) ← {}
∀ t_i ∈ **R**.getTransl(*t*), unmarked(t_j)
let sense *ts* ← { t_i }
mark(t_i);
// lookahead
∀ t_j ∈ **R**.getTransl(*t*), unmarked(t_j) |
∃ t_{ts} ∈ terms(*ts*) :
{**R**.getTransl(t_j) \ *t*} ∩ {**R**.getTransl(t_{ts}) \ *t*} = ∅
do { **let** *ts* ← *ts* ∪{ t_j }; mark(t_j) }
// lookback
If { ∃ *s* ∈ senses(*t*), t_s ∈ terms(*s*), t_{ts} ∈ terms(*ts*),
t_{sk} ∈ **R**.getTransl(t_s) |
t_{sk} ∈ **R**.getTransl(t_{ts})
Then let *s* ← *s* ∪ *ts*
Else senses(*t*) ← senses(*t*) ∪ { *ts* }
EndIf
Return senses(*t*)

Figure 4: algorithm for term clustering, bilingual

term	t1s	t2s
imposta	edition	edizione
		emissione
		~~imposta~~
	window-shutter	persiana
		~~imposta~~
	shutter	~~imposta~~
		serranda
		persiana
	tax	~~imposta~~
		tassa
		daziare
		erariale

Table 2: Freelang English translations for word "Imposta

Our task is equivalent to partitioning the graph so that every vertex in a partition is connected (has a path) to all the other vertices in the same partition (this is not a *clique*, since we are considering even paths connecting two vertices through more than an edge) and is not connected to any other vertices in the graph.

3.2. Term clustering, bilingual resources

The algorithm for clustering translations of bilingual dictionaries (figure 4) is slightly different, because terms in <ti> (i.e. translations of the IndexTerm), are not comparable with terms in <t2> (reverse translations), as they belong to different idioms. For this reason, the algorithm no more searches for couples of terms from <t1> and <t2>, but for set overlaps among terms in <t2>. Returning back to the graph morphism, the third statement becomes:

- a edge/*t2* connects two vertices/*t1s* if *t2* is a translation for both the *t1s*

The next example shows algorithm execution for English translations, again of the Italian word "imposta". For this example we adopted the English-Italian bilingual dictionary from the set of Freelang [OR3] bilingual dictionaries:

senses(*Imposta*) = {}

examining: *edition* (translations: *edizione*, *emissione*):
no matches neither in lookahead nor in lookback
senses(*imposta*) = {{*edition*}}

examining: *window-shutter* (translations: *persiana*):
lookahead: *persiana* ∈ **R**.getTransl(*shutter*) **then**:
ts ← {*window-shutter, shutter*}; mark(*shutter*);
senses(*imposta*) = {{*edition*},{*window-shutter, shutter*}}

examining: *tax*:
no matches neither in lookahead nor in lookback

senses(*imposta*) = {{*edition*},{*window-shutter*, *shutter*},{*tax*}}

Figure 5: step-by-step execution of term clustering algorithm for translations

4. Experimental Results

Extensive experimentation has been carried on a set of 60 ambiguous English words, running a tool based on the two algorithms which clustered synonyms and translations provided by two linguistic resources: WordNet (for synonyms) and a Freelang English-Italian dictionary (for translations). The two resources have been accessed by the tool through two implementations of the Linguistic Watermark (Pazienza & Stellato, 2006) library, a set of java interfaces and abstract classes for providing uniform access to heterogeneous LRs.

WordNet structure already implies senses separations for word definitions so, in our first experiment, for every term from the test set we simply collapsed the synonyms related to all of its senses, obtaining flat WORDLISTs. The aim of the algorithm was thus to rediscover the separations for terms in the WORDLISTs according to the original senses provided by WordNet. We then used WordNet original synset structure as an oracle for testing the soundness of our algorithm.

In the second experiment, we had to manually create an oracle, separating translations according to their different senses. What "sense" mean is obviously a partially subjective perception of the annotators creating the oracle, which may vary mostly on how fine is the grain upon which conceptual differences between very analogous terms are considered. Nonetheless, we reported an interannotator agreement of 96,7% on the whole test set, and let a third annotator take final decisions for the remaining 2,3%, thus guaranteeing as much objectivity as possible. We stress the fact that, through the oracle is humanly annotated, it is still strictly dependant on the specific resource from which it has been created, as no term has been added nor removed from the resource.

Table 3 reports the results of the experiments. To cope with this kind of set-oriented measures, the base unit upon which results have been computed is the number of possible pairs of terms which appear in the same sense. So, referring to our first example:

imposta = { {*sportello, anta, battente*},
{*persiana, scuro, scuretto* },
{*tassa*, *tributo*} }

imposta generates 3+3+1 different pairs. Precision and recall have thus been computed in the usual way, by comparing the number of matches in the oracle and in the senses produced by the algorithm.

Resource	Precision		Recall	
	Global	Average	Global	Average
Wordnet	52,9%	82,52%	99,48%	98,1%
Freelang	75%	88,54%	60,71%	77,7%

Table 3: Evaluation of the algorithms on WordNet and Freelang (English-Italian) resources

Global statistics have been taken considering the possible pairs of synonyms/translations as the unit of measurement, while averaged statistics first take local precision and recall values for every single ambiguous word and then average these results on the whole set of words. This second measure has been chosen to mitigate effects of resources which have a really dense sense-granularity. Suggestions produced for WordNet are in fact generally more than acceptable, but there are words, like "run", which expose nearby sixty different senses, often related to nuances of a few really different meanings. These words would have affected too much the global results, expecially considering the (relatively) small size of the test set. A more fair measure has thus been chosen to give an average idea of how good are suggestions, given whatsoever word as input.

5. Possible Applications

As anticipated, one immediate application of this algorithm is to automatically produce suggestions for developers building new linguistic resources from (a set of) existing ones. Flat entries, in the form of lists of translations/synonyms for searched terms, can be transformed to structured sets of words related to different senses. This approach can even be further extended to the production/extension of complex WordNet-like resources to new languages. Starting from a wordnet W_1 (for language α), and given a flat bilingual dictionary D (from language α to β), it is possible to build a new wordnet W_2 (for language β) by using a core synset structure taken from W_1 and automatically populating it with terms from D. A first trivial procedure, which exploits the term-clustering algorithm for translations, is the following:

buildNewWordnet(*wordnet* W_1,*flatRes D*) *wordnet* W_2
Begin
∀ $syn \in W_1$,
  let $terms_\beta(syn) \leftarrow \varnothing$
  ∀ $t\alpha_i \in syn$
    let *SET OF sense* $s_i\{\} \leftarrow$ **clusterTranslations**(t_i)
  ∀ $s_{ik} \in s_i\{\} \mid \exists\, s_{jn}, j \neq i$: $s_{ik} \cap s_{jn} \neq \varnothing$
    let $terms_\beta(syn) \leftarrow terms_\beta(syn) \cup s_{ik}$
End

Figure 6: suggested algorithm for generating new wordnets for different languages

Here, the use of the procedure **clusterTranslations** instead of mere translations guarantees a wider (and reliable) coverage of translated terms, whereas a get-all-translations approach would add spurious terms and an approach based on simple set-overlaps of translated terms would cut off singletons (terms in β which are translations for only one of the α-synonyms of a given synset) from the set of suggested terms. Obviously, a wordnet is not an ontology, its synset structure is organized in function of the specific idiom that must be represented, and should thus be changed when moving to a new language. However, a strict core of synset (as this has been the case for EuroWordNet approaches) may equally represent a first back bone of general linguistic knowledge, through which a new resource can be developed.

Another interesting application is *automatic linguistic enrichment of ontologies* (Pazienza & Stellato, 2005). A procedure which suggests synonyms and translations to the knowledge engineer and which is in grade of separating candidate terms according to their different meanings, could ease the task of ontology developers willing to better expose their semantic content in a linguistically motivated fashion.

6. Conclusions

In this work we have proposed two algorithms for clustering synonyms and translations of a given term, based on the analysis of their back translations. Experimental results have been reported on their application to two existing resources. The greatest limitation of these algorithms (which is probably intrinsic in the available information in the considered context) is related to the impossibility of finding different senses of a word which are completely identical under an extensional analysis (that is, they share the same set of synonyms, up to the trivial case of singletons bearing different interpretations). Possible applications have been discussed, which go from the most immediate objective of generating structured lexical resources, to supporting natural language documentation in advanced knowledge structures such as ontologies.

7. Cited Online Resources

[OR1] Babylon: www.babylon.com

[OR2] DICT: www.dict.org/bin/Dict

[OR3] Freelang: www.freelang.com

[OR4] WordNet: http://www.cogsci.princeton.edu/~wn/

8. References

Cole, R. A., Mariani, J., Uszkoreit, H., Zaenen, A., & Zue, V. (1997). *Survey of the State of the Art in Human Language Technology*. Cambridge, UK: Cambridge University Press.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. (1993). *Introduction to WordNet: An On-line Lexical Database.*

Pazienza, M. T., & Stellato, A. (2006). Exploiting Linguistic Resources for building linguistically motivated ontologies in the Semantic Web. *Second Workshop on Interfacing Ontologies and Lexical Resources for Semantic Web Technologies (OntoLex2006), held jointly with LREC2006.* Magazzini del Cotone Conference Center, Genoa, Italy.

Stamou, S., Oflazer, K., Pala, K., Christoudoulakis, D., Cristea, D., Tufiş, D., et al. (2002). BALKANET: A Multilingual Semantic Network for the Balkan Languages. *International Wordnet Conference*, (p. 12-14). Mysore, India.

Vossen, P. (1998). *EuroWordNet: A Multilingual Database with Lexical Semantic Networks.* Dordrecht: Kluwer Academic Publishers.