



HAL
open science

Inferring knowledge from a large semantic network

Dominique Dutoit, Thierry Poibeau

► **To cite this version:**

Dominique Dutoit, Thierry Poibeau. Inferring knowledge from a large semantic network. COLING 2002: The 19th International Conference on Computational Linguistics, Aug 2002, Taipei, Taiwan. pp.232-238. hal-00004061

HAL Id: hal-00004061

<https://hal.science/hal-00004061>

Submitted on 25 Jan 2005

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Inferring knowledge from a large semantic network

Dominique Dutoit

Memodata et CRISCO
17, rue Dumont d'Urville
F-14000 Caen

memodata@wanadoo.fr

Thierry Poibeau

Thales and LIPN
Domaine de Corbeville
F-91404 Orsay

thierry.poibeau@thalesgroup.com

Abstract

In this paper, we present a rich semantic network based on a differential analysis. We then detail implemented measures that take into account common and differential features between words. In a last section, we describe some industrial applications.

1 Introduction: textual and differential semantics

In textual analysis, each lexical item from a text is broken down in a list of semantic features. Features are intended to differentiate one word from another: a naive example would be a feature `back` that could express the difference between a chair and a stool. Of course, most of the time, features are not so easy to define. Some feature typologies have been provided, but there are still much discussions about the nature of a feature in a text. Most of the studies concerning differential semantics are based on a human approach to texts (this can lead to different problems, see below). Textual Semantics, also called differential semantics, is revisiting the concepts of continental structuralism like decomponential semantics (Cavazza, 1998).

The problem is then to have a lexical formalism that allows, for a lexical item, a simple description and some other features which could be dynamically inferred from the text. For example, the dictionary should mention that a “door” is an aperture, but it is more questionable to mention in the dictionary that “one can walk through a door”. However,

it can be an important point for the interpretation of a sentence in context.

That is the reason why Pustejovsky introduced in the nineties the notion of “generative lexicon” (Pustejovsky, 1991) (Pustejovsky, 1995). His analysis has to deal with the notion of context: he proposes to associate to a word a core semantic description (the fact that a “door” is an “aperture”) and to add some additional features, which can be activated in context (“walk-through” is the *telic role* of a “door”). However, Pustejovsky does not take into account important notions such as lexical chains and text coherence. He proposes an abstract model distant from real texts.

Semantic features can be used to check out text coherence through the notion of “isotopy”. This notion is “the recurrence within a given text section (regardless of sentence boundaries) of the same semantic feature through different words” (Cavazza, 1998). The recurrences of these features throughout a text allows to extract the topic of interest and some other points which are marginally tackled in the text. It provides interesting ways to glance at the text without a full reading of it; it also helps the interpretation.

In this paper, we present a rich semantic network based on a differential analysis. We then detail implemented measures that take into account common and differential features between words. In a last section, we describe some industrial applications.

2 The semantic network

The semantic network used in this experiment is a multilingual network providing information for 5 European languages. We quickly describe the

network and then give some detail about its overall structure.

2.1 Overall organisation

The semantic network we use is called *The Integral Dictionary*. This database is basically structured as a merging of three semantic models available for five languages. The maximal coverage is given for the French language, with 185.000 word-meanings encoded in the database. English Language appears like the second language in term of coverage with 79.000 word-meanings. Three additional languages (Spanish, Italian and German) are present for about 39.500 senses.

These smallest dictionaries, with universal identifiers to ensure the translation, define the Basic Multilingual Dictionary available from the ELRA. Grefenstette (1998) has done a corpus coverage evaluation for the Basic Multilingual Dictionary. The newspapers corpora defined by the US-government-sponsored Text Retrieval Conference (TREC, 2000) has been used as a test corpus. The result was that the chance of pulling a random noun out of the different corpus was on average 92%. This statistic is given for the Basic Multilingual Dictionary and, of course, the French Integral Dictionary reaches the highest coverage.

This semantic network is richer than Wordnet (Bagga *et al.*, 1997) (Fellbaum, 1998): it has got a larger number of links and is based on a componential lexical analysis. Because words are highly interconnected, the semantic network is easily tunable for a new corpus (see section 2.3).

2.2 Measure of distance between words

We propose an original way to measure the semantic proximity between two words. This measure takes into account the similarity between words (their common features) but also their differences.

Let's take the following example:

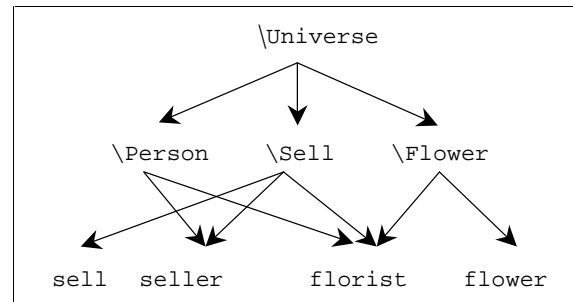


Figure 1: An example of semantic graph

The comparison between two words is based on the structure of the graph: the algorithm calculates a score taking into account the common ancestors but also the different ones. Let's take the example of *seller* and *florist*. They have two common ancestors: `\Person` and `\Sell`, but also one differential element: the concept `\Flower` that dominates *florist* but not *seller*.

The notion of “nearest common ancestor” is classical in graph theory. We extend this notion to distinguish between “symmetric nearest common ancestor” (direct common ancestor for both nodes) and “asymmetric nearest common ancestor” (common ancestor, indirect at least for one node).

Definition: Distance between two nodes in a graph

We note d the distance between two nodes A and B in a graph. This distance is equivalent to the number of intervals between two nodes A and B . We have $d(A, B) = d(B, A)$.

Example: We have $d(\text{sell}, \text{\Sell}) = 1$ and $d(\text{sell}, \text{\Universe}) = 2$, from Figure 1. Note that $d(\text{sell}, \text{\Sell}) = d(\text{\Sell}, \text{sell}) = 1$.

Given:

$h(f)$ = the set of ancestors of f .

$c(f)$ = the set of arcs between a daughter f and the graph's root.

We have:

$h(\text{seller}) = \{\text{\Sell}, \text{\Person}, \text{\Universe}\}$

$c(\text{seller}) = \{(\text{seller}, \text{\Sell}), (\text{seller}, \text{\Person}), (\text{\Sell}, \text{\Universe}), (\text{\Person}, \text{\Universe})\}$

etc.

Definition: Nearest common ancestors (NCA)

The nearest common ancestors between two words A and B are the set of nodes that are daughters of $c(A) \cap c(B)$ and that are not ancestors in $c(A) \cap c(B)$.

Example: From Figure 1, we have:

$$c(\text{seller}) \cap c(\text{florist}) = \{ \backslash\text{Sell}, \backslash\text{Universe}, (\backslash\text{Person}, \backslash\text{Universe}) \}$$

$$\text{DaughterNodes}(c(\text{seller}) \cap c(\text{florist})) = \{ \backslash\text{Sell}, \backslash\text{Person} \}$$

$$\text{AncestorNodes}(c(\text{seller}) \cap c(\text{florist})) = \{ \backslash\text{Universe} \}$$

The NCA is equal to the set of nodes in the set $\text{DaughterNodes}(c(\text{seller}) \cap c(\text{florist}))$ but not in $\text{AncestorNodes}(c(\text{seller}) \cap c(\text{florist}))$. Given that no element from $\text{AncestorNodes}(c(\text{seller}) \cap c(\text{florist}))$ appears in $\text{DaughterNodes}(c(\text{seller}) \cap c(\text{florist}))$, we have:

$$\text{NCA}(\text{seller}, \text{florist}) = \{ \backslash\text{Sell}, \backslash\text{Person} \}$$

We then propose a measure to calculate the similarity between two words. The measure is called *activation* and only takes into account the common features between two nodes in the graph. An equal weight is attributed to each NCA. This weight corresponds to the minimal distance between the NCA and each of the two concerned nodes.

Definition: activation (d_λ)

The *activation* measure d_λ is equal to the mean of the weight of each NCA calculated from A and B :

$$d_\lambda(A, B) = \frac{1}{n} \sum_{i=1}^n (d(A, \text{NCA}_i) + d(B, \text{NCA}_i))$$

The activation measure has the following properties:

- $d_\lambda(A, A) = 0$, because A is the unique NCA of A λ A.
- $d_\lambda(A, B) = d_\lambda(B, A)$ (symmetry)
- $d_\lambda(A, B) + d_\lambda(B, C) \geq d_\lambda(A, C)$ (euclidianity)

Example : According to Figure 1, we have $\text{NCA}(\text{seller}, \text{florist}) = \{ \backslash\text{Sell}, \backslash\text{Person} \}$. Consequently, if we assign a weight equal to 1 to each link, we have:

$$d_\lambda(\text{seller}, \text{florist}) = (d(\text{seller}, \backslash\text{Sell}) + d(\backslash\text{Sell}, \text{florist}) + d(\text{seller}, \backslash\text{Person}) + d(\backslash\text{Person}, \text{florist})) / 2$$

$$d_\lambda(\text{seller}, \text{florist}) = (1 + 1 + 1 + 1) / 2$$

$$d_\lambda(\text{seller}, \text{florist}) = 2$$

We can verify that:

$$d_\lambda(\text{florist}, \text{seller}) = d_\lambda(\text{seller}, \text{florist}) = 2$$

The set of NCA takes into account the common features between two nodes A et B. We then need another measure to take into account their differences. To be able to do that, we must define the notion of asymmetric nearest common ancestor.

Definition: Asymmetric nearest common ancestor (ANCA)

The asymmetric nearest common ancestors from a node A to a node B is contained into the set of ancestors of $c(B) \cap c(A)$ which have a direct node belonging to $h(A)$ but not to $h(B)$.

Example: According to Figure 1, we have:

$$\text{AncestorNodesNotNCA}(c(\text{seller}) \cap c(\text{florist})) = \{ \backslash\text{Universe} \}$$

The concept $\backslash\text{Universe}$ does not have any daughter that is a member of $h(\text{seller})$ but not of $h(\text{florist})$. As a consequence, we have:

$$\text{ANCA}(\text{seller}, \text{florist}) = \emptyset$$

On the other hand, the concept $\backslash\text{Universe}$ has a daughter $\backslash\text{Flower}$ that belongs to $h(\text{florist})$ but not to $h(\text{seller})$. As a consequence, we have:

$$\text{ANCA}(\text{florist}, \text{seller}) = \{ \backslash\text{Universe} \}$$

It is now possible to measure the distance between two words from their differences. A weight is allocated to each link going from node N_i , asymmetric nearest common ancestor, to A and B. The weight is equal to the length of the minimal length of the path going from A to N_i and from B to N_i .

Definition: proximity (d_{\perp})

The proximity measure takes into account the common features but also the differences between two elements A and B and is defined by the following function:

$$d_{\perp}(A, B) = d_{\wedge}(A, B) + \frac{1}{n} \sum_{i=1}^n (d(A, ANCA_i) + d(B, ANCA_i))$$

Because the set of ANCA from a node A to a node B is not the same as the one from a node B to a node A, the proximity measure has the following properties:

- $d_{\perp}(A, A) = 0$, because $ANCA(A, A) = \emptyset$.
- $d_{\perp}(A, B) \neq d_{\perp}(B, A)$ if the set of ANCA is not empty (antisymmetry)
- $d_{\perp}(A, B) + d_{\perp}(B, C) \geq d_{\perp}(A, C)$ (euclidianity)

The proximity measure is dependent from the structure of the network. However, one must notice that this measure is a relative one: if the semantic network evolves, all the proximity measures between nodes are changed but the relations between nodes can stay relatively stable (note that the graph presented on Figure 1 is extremely simplified: the real network is largely more connected).

Example: Let's calculate the semantic proximity between *seller* and *florist*: $d_{\perp}(\text{seller}, \text{florist})$. We will then be able to see that the proximity between *florist* and *seller* does not produce the same result (antisymmetry).

Given that $ANCA(\text{seller}, \text{florist}) = \emptyset$, the second element of the formula based on the set of ANCA is equal to 0. We then have:

$$\begin{aligned} d_{\perp}(\text{seller}, \text{florist}) &= d_{\wedge}(\text{seller}, \text{florist}) + 0 \\ d_{\perp}(\text{seller}, \text{florist}) &= 2 + 0 \\ d_{\perp}(\text{seller}, \text{florist}) &= 2 \end{aligned}$$

$ANCA(\text{seller}, \text{florist})$ is the set containing the concept $\backslash\text{Universe}$, because the concept $\backslash\text{Flower}$ is an ancestor of *florist* but not of *seller*. We then have:

$$\begin{aligned} d_{\perp}(\text{florist}, \text{seller}) &= d_{\wedge}(\text{florist}, \text{seller}) + (d(\text{seller}, \backslash\text{Universe}) + d(\backslash\text{Universe}, \text{florist})) / 1 \\ d_{\perp}(\text{florist}, \text{seller}) &= 2 + (2 + 2) / 1 \\ d_{\perp}(\text{florist}, \text{seller}) &= 6 \end{aligned}$$

To sum up, we have:

$$\begin{aligned} d_{\wedge}(\text{florist}, \text{seller}) &= 2 \\ d_{\wedge}(\text{seller}, \text{florist}) &= 2 \\ d_{\perp}(\text{seller}, \text{florist}) &= 2 \\ d_{\perp}(\text{florist}, \text{seller}) &= 6 \end{aligned}$$

The proximity measure discriminates *florist* from *seller*, whereas the activation measure is symmetric. The componential analysis of the semantic network reflects some weak semantic differences between words.

2.3 Link weighting

All the links in the semantic network are typed so that a weight can be allocated to each link, given its type. This mechanism allows to very precisely adapt the network to the task: one does not use the same weighting to perform lexical acquisition as to perform word-sense disambiguation. This characteristic makes the network highly adaptive and appropriate to explore some kind of lexical tuning.

3 Experiment and evaluation through an information filtering task

In this section we propose to evaluate the semantic network and the measures that have been implemented through a set of NLP applications related to information filtering. To help the end-user focus on relevant information in texts, it is necessary to provide filtering tools. The idea is that the end-user defines a "profile" describing his research interests (van Rijsbergen, 1979) (Voorhees, 1999).

A profile is a set of words, describing the user's domain of interest. Unfortunately the measures we have described are only concerned with simple words, not with set of words.

We first need to slightly modify the activation measure, so that it accepts to compare two sets of words, and not only two simple

words¹. We propose to aggregate the set of nodes in the graphs corresponding to the set of words in the profile. This node has the following properties:

$$h(M) = \bigcup_{i=1}^n h(m_i)$$

$$c(M) = \bigcup_{i=1}^n c(m_i)$$

where $h(M)$ is the set of ancestors of M and $c(M)$ the set of links between M and the root of the graph. It is then possible to compare two sets of words, and not only two simple words.

In the framework of an Information Extraction task, we want to filter texts to focus on sentences that are of possible interest for the extraction process (sentences that could allow to fill a given slot). We then need a very precise filtering process performing at the sentence level². We used the activation measure for the filtering task. A sentence is kept if the activation score between the filtering profile and the sentence is above a certain threshold (empirically defined by the end-user). A filtering profile is a set of words in relation with the domain or the slot to be fill, defined by the end-user.

We made a set of experiments on a French financial newswire corpus. The topic was the same as in the MUC-6 conference (1995): companies purchasing other companies. We made the experiment on a set of 100 news stories (no training phase).

The filtering profile was composed of the following words: *rachat*, *cession*, *enterprise* (buy, purchase, company). The corpus has been manually processed to identify relevant sentences (the reference corpus). We then compare the result of the filtering task with the reference corpus.

¹ This measure allows to compare two set of words, or two sentences. For a sentence, it is first necessary to delete empty words, to obtain a set of full words

² This is original since most of the systems so far are concern with texts filtering, not sentence filtering.

In the different experiments we made, we modified different parameters such as the filtering threshold (the percentage of sentences to be kept). We obtained the following results:

	10%	20%	30%	40%	50%
Precision	.72	.54	.41	.33	.28
Recall	.43	.64	.75	.81	.85

We also tried to normalize the corpus, that is to say to replace entities by their type, to improve the filtering process. We used a state-of-the-art named entity recogniser that was part of a larger toolbox for named entity recognition.

	10%	20%	30%	40%	50%
Precision	.75	.56	.43	.34	.29
Recall	.49	.71	.82	.89	.94

We notice that we obtain, from 10% of the corpus, a 75% precision ratio (3 sentences out of 4 are relevant) and nearly a 50% recall ratio. The main interest of this process is to help the end-user directly focus on relevant pieces of text. This strategy is very close from the EXDISCO system developed by R. Yangarber at NYU (2000), even if the algorithms we use are different.

4 Application services overview

In this section, we detail some of the applications developed from the semantic network described above. All of these applications are available through java API. They are part of the applicative part of the network called the Semiograph³. Most of the examples will be given in French.

4.1 Query expansion

This application gives a help to the users who query the web through a search engine. In this framework, the Semiograph has to determinate

³ Part of Speech tagging, syntactic analysis for French and Word Sense Disambiguation are also APIs of the Semiograph.

the sense of the query and generate (or suggest) an expansion of the query in accordance to the semantic and syntactic properties of the source.

The Semiograph links independent mechanisms of expansion defined by the user. Eight mechanisms are available :

- Alias: to get the graphics variant
- Synonyms: to get synonyms for a meaning
- Hypernyms: to get hypernyms for a meaning
- Hyponyms: to get hyponyms for a meaning
- Inflected forms : to get the inflected for a meaning
- Derived forms: to get correct lexical functions in accordance or not with the syntactical proposition
- Geographical belonging: to get toponyms
- Translation (language parameter) : to get a translation of the query.

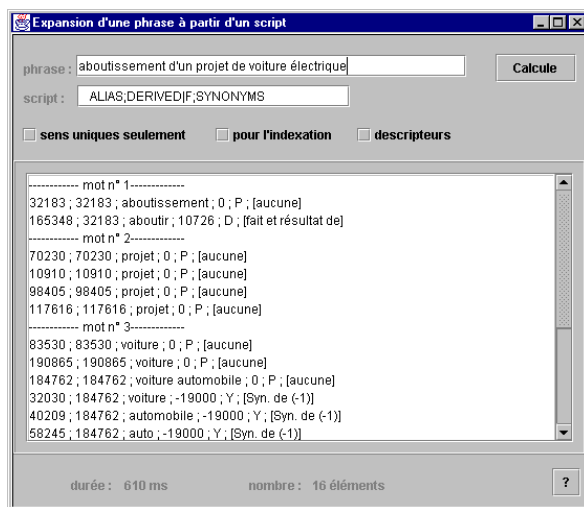


Figure 2: Query expansion

4.2 Word sense disambiguation and Term spotting

Lexical semantics provides an original approach for the term spotting task. Generally speaking, the main topics addressed by a document are expressed by ambiguous words. Most of the time, these words can be

disambiguated from the context. If a document treats of *billiards*, the context of *billiards* is necessarily saturated by terms of larger topics like *games*, *competition*, *dexterity*... and terms in dependence with *billiard* like *ball*, *cue*, *cannon*...

Using this property, lexical topics are found by measuring the semantic proximity of each plain word of a text with the text itself. Terms that have the minimal semantic proximity are the best descriptors.

Note that this property may be used to verify the relevance of keywords manually given by a writer. An application may be the struggle to the spamming of search engine. To give an example of result of lexical summary, the algorithm applied to this paper provides in the 20 best words the terms : *lexicon*, *dictionary*, *semantic network*, *semantics*, *measures* and *disambiguation*. All these terms are highly relevant.

4.3 Emails sorting and answering

In this application, we have to classify a flow of documents according to a set of existing profiles. Most systems execute this task after a learning phase. A learning phase causes a problem because it needs a costly preliminary manual tagging of documents. It is then attractive to see if a complex lexicon could perform an accurate classification without any learning phase.

In our experiments the end-user must have to define profiles that correspond to his domains of interest. The formalism is very light: firstly, we define an identifier for each profile; secondly we define a definition of this profile (a set of relevant terms according to the domain). On the following examples, identifiers are given between parentheses and definitions are given after.

[guerre du Kosovo] *guerre du Kosovo*
 [tabac et jeunesse] *tabac et jeunesse*
 [alcoolisme et Bretagne] *alcoolisme et Bretagne*
 [investissement immobilier en Ile-de-France] *achat, vente et marché immobilier en île-de-France*

The definitions may be given in English with the exactly same result. The following text :

Les loyers stagnent à Paris mais la baisse de la TVA sur les dépenses de réparation de l'habitat devrait soutenir le marché de l'ancien

gives in term of semantic proximity:

[guerre du Kosovo]	135
[tabac et jeunesse]	140
[alcoolisme et Bretagne]	129
[investissement immobilier en Ile-de-France]	9

We observe that differences between the mailboxes are very marked (the best score is the lowest one). Note that this approach may be used to help the classifying of web sites that is today entirely manually carry out.

5 Conclusion

In this paper, we have shown an efficient algorithm to semi-automatically acquire knowledge from a semantic network and a corpus. A set of basic services are also available through java APIs developed above the semantic network. We have shown that this set of elements offers a versatile toolbox for a large variety of NLP applications.

6 References

- Bagga A., Chai J.Y. et Biermann A. *The Role of WORDNET in the Creation of a Trainable Message Understanding System*. In Proceedings of the 14th National Conference on Artificial Intelligence and the Ninth Conference on the Innovative Applications of Artificial Intelligence (AAAI/IAAI'97), Rhode Island, 1997, pp. 941–948.
- Basili R., Catizone R., Pazienza M.T., Stevenson M., Velardi P., Vindigni M. and Wilks Y. (1998) An empirical approach to Lexical Tuning. Workshop on Adapting lexical and corpus resources to sublanguages and applications, LREC (Grenada).
- Cavazza M. (1998) Textual semantics and corpus-specific lexicons. Workshop on Adapting lexical and corpus resources to sublanguages and applications, LREC (Grenada).
- Fellbaum C. (1998) WordNet : An Electronic Lexical Database, edited by Fellbaum, M.I.T. press.

- Grefenstette G. (1998) Evaluating the adequacy of a multilingual transfer dictionary for the Cross Language Information Retrieval, LREC 1998.
- MUC-6 (1995) *Proceedings Sixth Message Understanding Conference (DARPA)*, Morgan Kaufmann Publishers, San Francisco.
- Pustejovsky J. (1991) The generative lexicon. *Computational Linguistics*, 17(4).
- Pustejovsky J. (1995) *The generative lexicon*, MIT Press, Cambridge.
- TREC (2000) The Ninth Text REtrieval Conference (TREC 9). Gaithersburg, 2000. http://trec.nist.gov/pubs/trec9/t9_proceedings.html.
- van Rijsbergen C.J. (1979) *Information Retrieval*. Butterworths, Londres.
- Voorhees, E.M. (1999) Natural language processing and information retrieval. In M.T. PAZIENZA (éd.), *Information extraction, toward scalable, adaptable systems*, Springer Verlag (Lecture Notes in computer Science), Heidelberg, pp. 32–48.
- Yangarber R. (2000) *Scenario Customization for Information Extraction*. PhD Thesis, New York University.