Word sense ranking based on semantic similarity and graph entropy

João Sousa LOPES [a], Sergio ALVAREZ-NAPAGAO [a] and
Javier VÁZQUEZ-SALCEDA [a]
[a] *KEMLg - Knowledge Engineering and Machine Learning Group*
UPC - Universitat Politècnica de Catalunya
Barcelona, Spain
{jsousalopes,salvarez,jvazquez}@lsi.upc.edu

Abstract. In this paper we propose a system for the recommendation of tagged pictures obtained from the Web. The system, driven by user feedback, executes an abductive reasoning (based on WordNet synset semantic relations) that is able to iteratively lead to new concepts which progressively represent the cognitive creative user state. Furthermore we design a selection mechanism to pick the most relevant abductive inferences by mixing a topological graph analysis together with a semantic similitude measure.

Keywords. semiotics, wordnet, tags, abduction, inference, semantics

Introduction

AI often lays on the study of human cognitive processes as a metaphor for novel models of automatic learning. Indeed, many of this systems pursuit the idea of reproducing some kind of human information acquisition and processing skills in order to develop autonomous actions on a rich and changeable environment. However, the paradigm of assisting human cognition by using artificial systems[1] changes the perspective, in the sense that researchers try to accommodate automatic processes with as human problem-solving capabilities. This combination relies not only on the difficulty to process a complete autonomous model of these human abilities but mainly because researchers have known that individual decision makers are often unable to make the best decisions when the problem is complex. Tversky and Kahneman [26] argue that decision makers are open to serious errors and biased decisions because they commonly use *rules of thumb* or heuristics.

In [11,12] we presented *USE*, a novel system that guides the user on a constructive and iterative ideation process by using their natural expertise, in order to identify contextual sign meaning association. Our approach is based on abductive reasoning, an inference scheme originally introduced by Peirce [20]. The standard formulation describes abduction as the inference of a hypothesis C that explains the evidence E, given the law $C \rightarrow E$. This form of abduction became a prevalent reasoning mechanism in many

[1] Often solved using what is known as decision support systems (DSS).

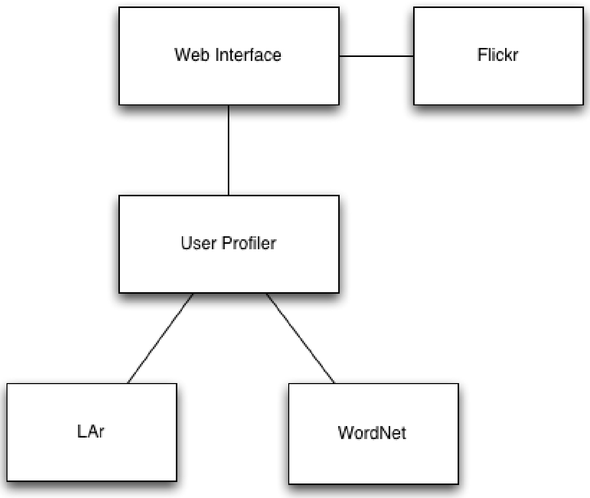

Figure 1. Architecture of *USE*

fields of artificial intelligence such as diagnosis, natural language understanding, default reasoning, database updates, planning, and high-level vision [17,8,21]. Our system aims to find the find the best image for the word-based user inputs through a specifically designed browser tool which at the same time, transparently, tries to minimise the efforts of connecting concepts that are semantically distant.

The process of brainstorming, as well as speech processing, is linear and opposed to the natural process of ideation which is parallel. Therefore, and because the computational cost of matching all possible parallel reasoning on a knowledge graph is NP complete, we need to design an heuristics methods that selects from the set of abductions those ones that play a more important role.

1. *USE*

USE [11] is an image recommender system intended for creative brainstorming. The architecture of *USE* is depicted in Figure 1.

The *Web Interface* is the link between *USE* and the user. In a Flash webpage, the user starts by typing a set of words. These words are used to retrieve a set of images from *Flickr*, each of which will be voted as positive or negative. After the images are given a value by the user, *USE* makes a new refined search based on the set of tags identifying each image in *Flickr*. This process is repeated until the user finds an appropriate image.

For the *Web Interface*, each time an input is provided by the user is seen as a *user interaction*. In the case of a new search, this interaction is defined as the set of words typed by the user. In the case of a refinement step, the result of the interaction is:

- For each picture marked as positive, all of their tags will be added as predicates.
- For each picture marked as negative, all of their tags that are not in any of the pictures marked as positive will be added as negated predicates, that is, with a truth value of false.

- For each picture left unmarked, nothing is done.

The main component of this architecture is the *User Profiler*, which receives all these user interactions and produces a new set of tags which will be used by the *Web Interface* as search terms for new images. In order to produce these new tags, the *User Profiler* uses two independent modules: *WordNet* and *LAr*.

LAr [13] is a formal adaptive logic for abduction (see Section 1.1). This module has already been implemented in Ruby. Examples of traces can be found in [12].

WordNet [15] is a large browseable lexical database, available in several languages, which groups synsets expressing a distinct concept. We use WordNet as a support tool for the retrieval of semantic knowledge associated to tags.

1.1. Abduction

Abduction, as opposed to *deduction* and *induction*[2], is based on the inference of ϕ (*explanans*) from knowledge of the rule $\phi \rightarrow \psi$ and the observation ψ (*explanandum*). This means that abduction is not an analytic form of inference, but rather based on the Affirming the Consequent fallacy. Like induction, abduction is defeasible: the arrival of new observations might invalidate prior abductive inferences.

Our main objective is to show how logic conclusions inferred via abductions can be useful to construct a knowledge model that can be reused in a knowledge retrieval process. We have already applied the concept of reasoning-driven abduction in a model of a real industrial process, and a trace of the abduction inferences can be found at [11].

The formal logic used in our proposal is $\mathbf{LA}^r$, presented in [13]. $\mathbf{LA}^r$ is a logic based on Classical Logic with a non-monotonic dynamical process in which deductive steps are combined with abductive steps. Abductive steps may be withdrawn if, via deductive steps, its negation is derived.

This logic represents *abductive steps* as formulas of the form:

$$B(\beta), (\forall\alpha)(A(\alpha) \supset B(\alpha))/A(\beta) \tag{1}$$

In this formula we identify three components: $B(\beta)$, which is the fact to be explained (*explanans*), $(\forall\alpha)(A(\alpha) \supset B(\alpha))$, which is the *deductive rule*, and $A(\beta)$, which is the explanation (*explananda*).

In WordNet [15], the vocabulary is seen as a set W of pairs (f, s) where f is a word and s is a sense of that word. WordNet defines a set of semantic relations between these senses. From this set, we will focus on the two relations that bidirectionally describe the *Is-A* hierarchy: $hypernyms(s_i)$, and $hyponym(s_i)$.

The facts in our system are the predicates representing the user interaction. The rules are the semantic relations between words, in our case the ones representing the *Is-A* hierarchy: *hypernyms* and *hyponyms*. Considering a sense s, each of its hypernyms s' is a generalization, and thus $s \rightarrow s'$, and each of its hyponyms s'' is an specialization, and thus $s'' \rightarrow s$:

$$s' \in hypernym(s) \Rightarrow (\forall\alpha)(s(\alpha) \supset s'(\alpha) \wedge \neg s'(\alpha) \supset \neg s(\alpha)) \tag{2}$$

$$s'' \in hyponym(s) \Rightarrow (\forall\alpha)(s''(\alpha) \supset s(\alpha) \wedge \neg s(\alpha) \supset \neg s''(\alpha)) \tag{3}$$

---

[2]Deduction is based on the *modus ponens* syllogism ($\{\phi, \phi \rightarrow \psi\} \models \psi$), while induction is based on the inference of $\phi \rightarrow \psi$ as a rule from the observation of ϕ followed by ψ

We define the *synset graph* SG as a graph $SG =< V, E >$ where V is the set of noun synsets of WordNet and $\forall(u, v), u \in hypernyms(v) \vee u \in hyponyms(v) \rightarrow (u, v) \in E$. That is, the *synset graph* is composed of the senses of all the nouns in WordNet and there is an edge from each vertex towards all of its hypernyms and hyponyms.

All the rules are implicitly derived from the WordNet database at runtime and added to the proof, as needed.[3] At the beginning of the process, the proof is filled by the set of premises representing the senses of the word inputs of the user, and the rules derived from the senses of the main concept.

After each step of user interaction, an abduction process is executed in the proof and repeated until no new facts are added. Every time a contradiction appears in the proof, the conflicting facts and the abductions that originated them are pruned. That means that at every point in time, the proof is consistent but defeasible. A broader and more detailed explanation of the abduction process including an example can be found at [11].

2. Selection of *critical* nodes: word ranking

One of the most important properties of $\mathbf{LA}^r$ is that the abduction process is not bounded. That is, everything that can be abducted will be abducted, and the only way to ensure the *credibility* of the abductions is by the definition of *abnormality conditions* [13].

We proposed in [11] the use of a *post-abduction* verification mechanism to be carried out by the *User Profiler*. This verification mechanism is based on the theoretical concept of *criticality*, a measure of how important a particular abduction is, in terms of the reliability of the posterior inferences not only of its abduction, but more notably, of the whole set of inferences recursively inferred from it. The higher the number of inferences depend on an abduction, the more dangerous it is to work with them and the more fragile the posterior inferences are.

In other words, taking WordNet as a graph of words connected by their hierarchy defined by the *Is-A* relationship, we are looking for those words that are not only highly connected in terms of direct relationship with other words, but also in terms of semantic value.

Many approaches have been presented in order to tackle this problem, but they fall under only one of these categories: either they are focused on the topology of the graph ([24,16]), or they are focused on the semantics of the words: ([23,1,14]).

In this paper we present a mixed approach that creates a word sense ranking in a specific set[4], based on both their connectivity in the graph and their semantic relationship with respect to other words.

2.1. Graph connectivity measures

The measures for the selection of relevant word senses in a graph, based on the graph structure, are many [16] and can be classified as *local* or *global*.

[3] That is, whenever a word that has not yet been queried in WordNet appears.

[4] In our case, the whole set of noun synsets in WordNet, for a total of 74401.

2.1.1. *Local measures*

Local measures determine the relevance of a single node in terms of influence over the rest of the graph and thus are only useful when dealing with small graphs. The most important local measures are:

- In-degree Centrality: measures the number of edges terminating in a given vertex.
- Eigenvector Centrality: a more sophisticated version of degree centrality in which not all connections are equal. It assigns relative scores to all nodes in the graph based on the principle that connections to nodes having a high score contribute more to the score of the node in question [3,18,9].
- Key Player Problem: a vertex is considered important if it is relatively close to all other vertices [4].
- Betweenness Centrality: the fraction of shortest paths between node pairs that pass through a given vertex [7].

2.1.2. *Global measures*

Global measures, on the other hand, deal with the structure of the graph as a whole:

- Compactness: when compactness is high, each vertex can be easily reached from other vertices [5].
- Graph entropy: measures the amount of information and uncertainty in a random variable. When the entropy is high, many vertices are equally important, and when it is low, only a few vertices are relevant [25].
- Edge density: simply measures the ratio of edges in a graph over the number of edges of a complete graph with the same number of vertices.

2.1.3. *Graph entropy*

From all the connectivity measures seen in the previous sections, graph entropy is the only one that does not solely depend on the graph structure, but also on a probability distribution defined over the graph. This allows us to use semantic measures as the variable for the computation of graph entropy.

When used as a pure connectivity measure, graph entropy is defined as:

$$H(G) = \sum_{v \in V} p(v) \cdot log(\frac{1}{p(v)})$$

where:

$$p(v) = \frac{index(v)}{2|E|}$$

That is, every connection of each vertex is considered as equal.

2.2. *Semantic similarity measures*

WordNet has been throughfully been used as a test case for similarity measures [23, 1,14]. Semantic relatedness is a huge field of research inside NLP[5] and many are the

[5] Natural Language Processing.

algorithms that implement a specific unique measure. However, not many of them are already available as open source and that was a constraint for us, considering that the implementation of a semantic relatedness algorithm is out of the scope of our research.

The most complete suite for semantic relatedness computation is *Wordnet::Similarity* [19], a Perl library implementing as open source a set of the most important algorithms: Leacock & Chodorow (1998), Jiang & Conrath (1997), Resnik (1995), Lin (1998), Hirst & St-Onge (1998), Wu & Palmer (1994), the extended gloss overlap measure by Banerjee and Pedersen (2002), and two measures based on context vectors by Patwardhan (2003).

The algorithms implemented in *Wordnet::Similarity* score fairly well [2], but with respect to semantic relatedness algorithms in general, there is no consensus in the literature on how they should be ranked [22,6]. The fact is that their scoring depends completely on the test cases they are checked against, or on the set of words we take as dictionary, among many other factors.

Our intention is to use each one of these algorithms for the word ranking computation, but as a first test, we have decided to use the Extended Gloss Overlap Measure [2] based on the Lesk algorithm [10].

3. Graph entropy based on semantic similarity

Our proposal for a word sense ranking measure is therefore based on a combination of graph entropy and semantic relatedness. Graph entropy, when used in word sense disambiguation, does not consider any difference between the relationships between words, as previously seen.

Following our example, we will use the *synset graph* as the graph to be measured. However, with the graph as it is, there are only semantic relationships explicitly defined, all of them equal *a priori*.

All of the semantic relatedness algorithms are functions that compute a numerical value from two input word senses:

$$f : w \times w \rightarrow \mathbb{R}$$

Also, another common property of them is that $f(w_1, w_2) = f(w_2, w_1)$.

We will apply this function f on the edges, thus giving each edge a weight representing an objective value for the semantic similarity between the two word senses represented by its two vertices. We will define our *weighted synset graph* WSG of $SG =< V, E >$ as:

$$WSG =< V, E, f >$$

where V and E are equal to V and E in SG, and f is the semantic relatedness function.

In order to use graph entropy, we have to create a probability distribution on our graph. In our context of abduction, a word sense is *more likely* to be abducted when, compared to the rest of the word senses of the graph, it is more probable that a random word sense is semantically similar and with a direct relationship to the given one. For each node, this *likelihood* is then represented by the sum of the weights of its edges, divided by the total sum of weighs of all the edges in the graph:

$$likelihood(v) = \frac{\sum_{(v,w)\in E} f(v,w)}{\sum_{(x,y)\in E} f(x,y)}$$

A trivial consequence of this formula is that $\sum_{v\in V} likelihood(v) = 1$, so we can consider $likelihood()$ as the probability distribution for our graph entropy. The formula for our graph entropy will then be:

$$H(G) = \sum_{v\in V} likelihood(v) \cdot log\left(\frac{1}{likelihood(v)}\right)$$

Applied to WSG, this formula will give us the graph entropy for our test case.

Our objective, as seen previously, is to rank word senses by how relevant they are, but graph entropy as a measure will only output a global relevancy value for the whole graph, not for the vertices individually. However, graph entropy can also be used in an iterative procedure for the discovery of relevant nodes.

According to [24], the most important, or *relevant* for the purpose of our research, nodes are those which have the largest effect on the graph entropy when they are removed from the graph. It is more an empirical idea than an analytical one, but it has been proven to yield accurate results in other scenarios and it is our objective to test them in our test case. The intuition of the idea is that those nodes that belong to more abduction paths are relevant and, in addition, also those nodes that belong to unusual paths are relevant as well.

The pseudocode for the discovery of relevant nodes is [24]:

1. Compute the graph entropy
2. For all nodes v do the following:
   (a) Compute the entropy of v by calculating the entropy of all its edges as $E(v)$
   (b) Drop v from the graph
   (c) Calculate the entropy of the remnant graph as $EN(v)$
   (d) Calculate the cross entropy of $EN(v)$ and $E(v)$:
      - $Effect(v) = log\left(\frac{EN(v)}{E(v)}\right)$
3. Rank nodes based on $Effect(v)$

4. Practical results

The word sense ranking mechanism has been implemented in Perl according to the methods described in the previous sections (see Figure 2). This software is intended to be independent to *USE*, as its purpose is to precompute the word sense ranking and store it in a format available to the brainstorm application.

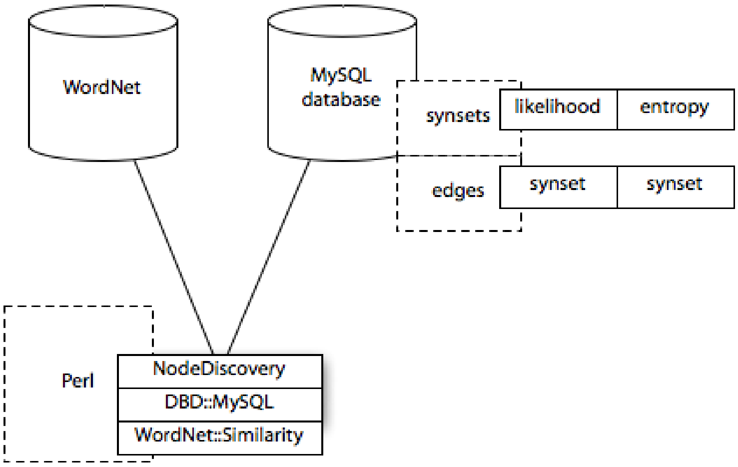

Figure 2. Deployment of the word sense ranking mechanism

abasia#n#1	disability_of_walking#n#1	98
disability_of_walking#n#1	abasia#n#1	98
abatable_nuisance#n#1	nuisance#n#1	680
nuisance#n#1	abatable_nuisance#n#1	680
abatement#n#2	abatement_of_a_nuisance#n#1	324
abatement_of_a_nuisance#n#1	abatement#n#2	324
abatement#n#2	moderation#n#4	107
moderation#n#4	abatement#n#2	107
abatement_of_a_nuisance#n#1	asbestos_abatement#n#1	369
asbestos_abatement#n#1	abatement_of_a_nuisance#n#1	369
abator#n#1	person#n#1	71
person#n#1	abator#n#1	71
abattis#n#1	line_of_defense#n#2	862
line_of_defense#n#2	abattis#n#1	862

Table 1. Hypernyms and hyponyms

4.1. Creation of the graph SG

The first step in the process has been to generate the graph SG based on WordNet. Starting by the top level noun word sense, *entity*, the program recursively searched for hypernyms, hyponyms and meronyms and recorded all of them in a *synset* table. The hypernym and hyponym relationships were recorded in a separate table *edges*. The total amount of records in the *synset* table is 74401, and the total amount of records in the *edges* table is 151700.

4.2. Likelihood computation

In order to get the WSG graph, we needed to add the likelihood value for each edge. The first step was to iteratively compute the result of the Lesk algorithm from *WordNet::Similarity* (see Table 1)

herb#n#1	0.005177202559444
tree#n#1	0.00279865173437837
class#n#7	0.00270995844622832
shrub#n#1	0.00263541211869066
asterid_dicot_genus#n#1	0.00258965609696064
dicot_genus#n#1	0.00243067120626914
monocot_genus#n#1	0.00166503344320215
dilleniid_dicot_genus#n#1	0.00162158206417841
rosid_dicot_genus#n#1	0.00152838585874072
symptom#n#1	0.00145238583349601
metallic_element#n#1	0.00141012223186705
animal_order#n#1	0.0013333599017272
chemical_element#n#1	0.00130712243091727
person#n#1	0.00130343500258723
orchid#n#1	0.00123025373265251

Table 2. Top 15 synsets by likelihood

The likelihood for each synset was then computed following the formula seen in Section 3:

$$likelihood(v) = \frac{\sum_{(v,w)\in E} f(v,w)}{\sum_{(x,y)\in E} f(x,y)}$$

Table 2 shows the top 15 synsets ordered by likelihood. A careful look at this results brings up a curious but logical conclusion: the synsets with a higher likelihood are *classifiers*. However, we expect this to change substantially after calculating the vertices entropy.

4.3. Graph entropy

Although the graph entropy of the full graph will not be needed for further experiments, it is a simple computation and it was done following the formula:

$$H(G) = \sum_{v\in V} likelihood(v) \cdot log\left(\frac{1}{likelihood(v)}\right)$$

The result was 0.91208681280251, which is a very high entropy. The interpretation of this result is that there is a high amount of synsets that are almost equally important.

4.4. Graph entropy per vertex

This is the last step prior to the discovery of relevant nodes and thus to the word sense final ranking. This is a hugely time consuming process, and due to time limitations, we have not been able to produce complete results yet. However, we have over 75 results that can be used to make an early analysis.

Table 3 is an extract of the top 10 synsets ordered by entropy at the moment. It must be noted that the order of the likelihood is almost the same as the entropy, but there are two notable exceptions: *White* and *Black* are higher in the list than they would be, should the list be ordered by likelihood.

synset	likelihood	entropy
body_of_water#n#1	0.000335609162096442	0.914167873638102
man#n#1	0.000266239416637492	0.913798930400534
woman#n#1	0.000238034135516821	0.913788207088082
Asian#n#1	0.00022103296355282	0.913735110905645
White#n#1	4.99930187054021e-05	0.913237186379223
Black#n#5	5.8396100572906e-05	0.913207856810315
Arab#n#1	0.00015588248704844	0.912950543614792
country#n#2	0.000127960854260848	0.912849233200891
physical_entity#n#1	2.70175037258982e-05	0.912743107505078
Semite#n#1	8.6034085027417e-05	0.91268699662846

Table 3. Synset entropies

The interpretation of this fact is that, although *Arab* and *country* have a higher total amount of semantic similarity with their neighbours, *White* and *Black* have a higher effect in the hierarchical paths that cross through them.

Seeing that we have only produced around 75 synset entropies at the moment, we expect to have interesting results when the whole process is finished.

5. Conclusions

In this paper we presented a novel WordNet based semantic selector that puts together a topological approach (based on a graph entropy computation) with a semantic weighted distributed probability. We have discussed the approach by unveiling a group of semantic measures, as well as others strictly restrain to the structure analysis. Afterwards, we selected an entropy-based measurement because of its featured combination of graph structure and probability distribution analysis over the graph. We then used a local semantic assess to calculate this probability, enabling consequently the entropy calculation.

Following this, we described the procedure used to measure the relative importance of each node by the effect of its removal over the global graph entropy.

This metrics are specially suited to accomplish the selection of abductive inferences over WordNet in an ideation process because the applied adaptive abduction logic, although has recursive coherence detection, does not elucidates what should be the best inferences, that if rebuked, would provide more information and, consequently, a more fine tune to potential future new abductions.

Abductive inference is often considered as a âĂŸcreativeâĂŹ type of reasoning because its hypothesis uncertainty by applying a known fallacy known as âĂŸAffirming the ConsequenceâĂŹ. For that reason we also described the architecture of a brainstorming system that uses the rated results of these abductions for suggesting new semantic relations by employing an interactive interface that present these inferences to ser approval.

The presented fallout, even being generated from partial results, showed promissory conclusions and the possible extension of these metrics for contextual knowledge retrieval and contextual meaning association. The presented paper is a crucial part of an extended framework of creative assistant tools designed to help humans to develop a conceptual search based on the semioticâĂŹs theory on sign processing and acquisition.

Acknowledgements

S. Alvarez-Napagao, J. Vázquez-Salceda would like to acknowledge support from European funded CONTRACT Project (IST-034418). J. Sousa Lopes work has been funded by FCT-Portugal, scholarship SFRH/BD/38767/2007. J. Vázquez-Salceda work has been also partially funded by the "Ramón y Cajal" program of the Spanish Ministry of Education and Science. The content of this paper however solely reflects the opinion of the authors, and does not necessarily represent the views of the European Commission.

References

[1] E. Agirre and G. Rigau. A proposal for word sense disambiguation using conceptual distance. *Arxiv preprint cmp-lg*, Jan 1995.

[2] S. Banerjee and T. Pedersen. Extended gloss overlaps as a measure of semantic relatedness. *International Joint Conference on Artificial Intelligence*, 18:805–810, 2003.

[3] E. Bonacich, E. Bonacich, E. Bonacich, and E. Bonacich. A theory of middleman minorities. 1973.

[4] S. Borgatti. Centrality and network flow. *Social Networks*, Jan 2005.

[5] R. Botafogo, E. Rivlin, and B. Shneiderman. Structural analysis of hypertexts: Identifying hierarchies and useful metrics. *ACM Transactions on Information Systems (TOIS)*, Jan 1992.

[6] I. Cramer. How well do semantic relatedness measures perform? a meta-study. *aclweb.org*.

[7] L. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, Jan 1977.

[8] M. Ishizuka and Y. Matsuo. Sl method for computing a near-optimal solution using linear and non-linear programming in cost- .... *Knowledge-Based Systems*, Jan 2002.

[9] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, Jan 1999.

[10] M. Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an .... *Proceedings of the 5th annual international conference on ...*, Jan 1986.

[11] J. S. Lopes, S. Alvarez-Napagao, S. Reis, and J. Vazquez-Salceda. Reasoning about abductive inferences in bdi agents. *Technical Report LSI-09-12-R, Universitat Politècnica de Catalunya*, pages 1–1, Apr 2009.

[12] J. S. Lopes, S. Alvarez-Napagao, and J. Vázquez-Salceda. Use: a concept-based recommendation system to support creative search. pages 1–5, May 2009.

[13] J. Meheus and D. Batens. A formal logic for abductive reasoning. *Logic Journal of IGPL*, Jan 2006.

[14] I. Melamed. Measuring semantic entropy. *ACL-SIGLEX Workshop Tagging Text with Lexical Semantics: Why ...*, Jan 1997.

[15] G. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, Jan 1995.

[16] R. Navigli and M. Lapata. Graph connectivity measures for unsupervised word sense disambiguation. *Proceedings of the International Joint Conference on Artificial Intelligence, Hyderabad, India*, 2007.

[17] Y. Ohsawa and M. Ishizuka. Networked bubble propagation method as a polynomial-timehypothetical reasoning for computing quasi-optimal solution. *Tools with Artificial Intelligence, 1993. TAI'93. Proceedings., Fifth International Conference on*, pages 184–187, 1993.

[18] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. *cpe.ku.ac.th*, Jan 1998.

[19] T. Pedersen, S. Patwardhan, and J. Michelizzi. Wordnet:: similarity-measuring the relatedness of concepts. *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, pages 1024–1025, 2004.

[20] C. Peirce and J. Buchler. The philosophy of peirce: selected writings. *books.google.com*, Jan 2000.

[21] H. Prendinger and M. Ishizuka. Qualifying the expressivity/eiticiency tradeoff: Reformation-based diagnosis.

[22] F. M. Pucher. Performance evaluation of wordnet-based semantic relatedness measures for word prediction in .... *userver.ftw.at*.

[23] R. Richardson, A. Smeaton, and J. Murphy. Using wordnet as a knowledge base for measuring semantic similarity between words. *Proceedings of AICS Conference. Trinity College*, Jan 1994.

[24] J. Shetty and J. Adibi. Discovering important nodes through graph entropy the case of enron email database. *Proceedings of the 3rd international workshop on Link discovery*, pages 74–81, 2005.

[25] G. Simonyi. Graph entropy: a survey. *Combinatorial Optimization: Papers from the Dimacs Special Year*, page 399, 1995.

[26] A. Tversky and D. Kahneman. Judgment under uncertainty: Heuristics and biases. *Science*, Jan 1974.