Proceedings of the

EACL 2009 Workshop on

GEMS: GEometrical Models of Natural Language Semantics

Endorsed by

the Association for Computational Linguistics
SIGLEX and SIGSEM, two Special Interest Groups of ACL

Edited by

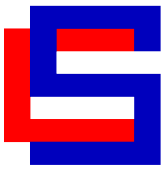
Roberto Basili
and
Marco Pennacchiotti

31 March 2009
Megaron Athens International Conference Centre
Athens, Greece

Production and Manufacturing by
TEHNOGRAFIA DIGITAL PRESS
7 Ektoros Street
152 35 Vrilissia
Athens, Greece

Endorsed by:

ACL SIGLEX



ACL SIGSEM

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

Introduction

The geometry of distributional models of lexical semantics represent a core topic in contemporary computational linguistics for its impact on several advanced Natural Language Processing tasks and some related knowledge fields (as social science and humanities).

The goal of the EACL 2009 GEMS Workshop on "*GEometrical Models of natural Language Semantics*" was to stimulate research on semantic spaces and distributional methods in NLP, by adopting an interdisciplinary view. This aimed to enforce the proper exchange of ideas, results and resources among often independent communities. The workshop provided a common ground for a fruitful discussion among experts of distributional approaches, collocational corpus analysis and machine learning, researchers interested in the use of quantitative models in NLP applications (like question answering, summarization or textual entailment), experts in formal computational semantics and in other fields of science as well.

The workshop successfully gathered a relevant number of high quality contributions to problems of meaning representation, acquisition and use, based on distributional and vector space models. We received 21 submissions, including short and long papers. Long papers were peer-reviewed by three members of the program committee, short papers by two. As an outcome of the review process, the program committee selected 11 papers for a full presentation, and 4 for short ones. All selected paper have been included in these proceedings. The paper are representative of the current state of the art in the subject, including:

- cutting edge researches on geometric methods and machine learning, such as tensor factorization, kernel methods and Dirichlet process mixture models;
- applications of semantic space models to NLP tasks, such as Textual Entailment Recognition, Ontology Learning, Induction of Selectional Preferences, Verb Classification and Machine Translation
- novel uses of distributional methods for advanced linguistic studies, such as lexical variation and evolution as well as for educational purposes;
- reference comparative studies among different types of semantic spaces.

The papers included in this volume shed some light on the state of the art and the potential applications of semantic spaces in NLP and in related linguistic fields.

We would like to thank all the authors for their hard work dedicated to the submissions. Our deepest gratitude goes to the members of the program committee for their precious reviewing. Most of the impact of this volume is entirely due to their careful analysis and meaningful suggestions to the authors. A special thank goes to Patrick Pantel for his stimulating and visionary invited talk, supported by his own institution. Finally, we acknowledge the EACL 2009 workshop chairs, Miriam Butt, Stephan Clark as well as Kemal Oflazer and David Schlangen, for their constant support across all the preparatory work.

Roberto Basili, University of Roma, *Tor Vergata*, Italy
Marco Pennacchiotti, Yahoo! Inc, Santa Clara, US.

March, 2009

Organizers:

Roberto Basili, University of Roma *Tor Vergata* (Italy)
Marco Pennacchiotti, Yahoo! Inc., Santa Clara, CA (US)

Program Committee:

Marco Baroni, University of Trento (Italy)
Michael W. Berry, University of Tenneesee (US)
Gemma Boleda, Pompeu Fabra University of Barcelona (Spain)
Johan Bos, University of Roma "*La Sapienza*" (Italy)
Paul Buitelaar, DFKI (Germany)
John A. Bullinaria, University of Birmingham (UK)
Rodolfo Delmonte, University *Ca' Foscari* Venice (Italy)
Katrin Erk, University of Texas (US)
Stefan Evert, University of Osnabruck (Germany)
Alfio Massimiliano Gliozzo, STLab - ISTC-CNR (Italy)
Jerry Hobbs, University of Southern California (US)
Alessandro Lenci, University of Pisa (Italy)
Jussi Karlgren, Swedish Institute of Computer Science (Sweden)
Will Lowe, University of Nottingham (UK)
Diana McCarthy, University of Sussex (UK)
Alessandro Moschitti, University of Trento (Italy)
Saif Mohammad, University of Maryland (US)
Sebastian Padó, Stanford University (US)
Patrick Pantel, Yahoo! Inc. (US)
Massimo Poesio, University of Trento (Italy)
Magnus Sahlgren, Swedish Institute of Computer Science (Sweden)
Sabine Schulte im Walde, University of Stuttgart (Germany)
Hinrich Schütze, University of Stuttgart (Germany)
Fabrizio Sebastiani, CNR (Italy)
Suzanne Stevenson, University of Toronto (Canada)
Peter D. Turney, National Research Council (Canada)
Dominic Widdows, Google Research (US)
Yorick Wilks, University of Sheffield (UK)
Fabio Massimo Zanzotto, University of Roma "*Tor Vergata*" (Italy)

Table of Contents

Conference Program

Tuesday, March 31, 2009

8:45–9:00 Opening Remarks

9:00–10:00 Invited Talk by Patrick Pantel

Session 1

9:50–10:15 *One Distributional Memory, Many Semantic Spaces*
Marco Baroni and Alessandro Lenci

10:15–10:40 *Word Space Models of Lexical Variation*
Yves Peirsman and Dirk Speelman

10:40–11:00 Coffee break

Session 2

11:00–11:25 *Unsupervised Classification with Dependency Based Word Spaces*
Klaus Rothenhäusler and Hinrich Schütze

11:25–11:50 *A Study of Convolution Tree Kernel with Local Alignment*
Lidan Zhang and Kwok-Ping Chan

11:50–12:15 *BagPack: A General Framework to Represent Semantic Relations*
Amaç Herdağdelen and Marco Baroni

Tuesday, March 31, 2009 (continued)

Short Presentations

12:15–12:30 *Positioning for Conceptual Development using Latent Semantic Analysis*
Fridolin Wild, Bernhard Hoisl and Gaston Burek

12:30–12:45 *Semantic Similarity of Distractors in Multiple-Choice Tests: Extrinsic Evaluation*
Ruslan Mitkov, Le An Ha, Andrea Varga and Luz Rello

12:45–13:45 Lunch break

Session 3

13:45–14:10 *Paraphrase Assessment in Structured Vector Space: Exploring Parameters and Datasets*
Katrin Erk and Sebastian Padó

14:10–14:35 *SVD Feature Selection for Probabilistic Taxonomy Learning*
Francesca Fallucchi and Fabio Massimo Zanzotto

14:35–15:00 *Unsupervised and Constrained Dirichlet Process Mixture Models for Verb Clustering*
Andreas Vlachos, Anna Korhonen and Zoubin Ghahramani

15:00–15:25 *A Non-negative Tensor Factorization Model for Selectional Preference Induction*
Tim Van de Cruys

Short Presentations

15:25–15:40 *A Graph-Theoretic Algorithm for Automatic Extension of Translation Lexicons*
Beate Dorow, Florian Laws, Lukas Michelbacher, Christian Scheible and Jason Utt

15:40–15:55 *Handling Sparsity for Verb Noun MWE Token Classification*
Mona Diab and Madhav Krishna

16:00–16:30 Coffee break

Tuesday, March 31, 2009 (continued)

Session 4

16:30–16:55 *Semantic Density Analysis: Comparing Word Meaning across Time and Phonetic Space*
Eyal Sagi, Stefan Kaufmann and Brady Clark

16:55–17:20 *Context-theoretic Semantics for Natural Language: an Overview*
Daoud Clarke

Panel

17:20–18:00 Discussion and Concluding Remarks

One distributional memory, many semantic spaces

Marco Baroni
University of Trento
Trento, Italy
marco.baroni@unitn.it

Alessandro Lenci
University of Pisa
Pisa, Italy
alessandro.lenci@ilc.cnr.it

Abstract

We propose an approach to corpus-based semantics, inspired by cognitive science, in which different semantic tasks are tackled using the same underlying repository of distributional information, collected once and for all from the source corpus. Task-specific semantic spaces are then built on demand from the repository. A straightforward implementation of our proposal achieves state-of-the-art performance on a number of unrelated tasks.

1 Introduction

Corpus-derived distributional *semantic spaces* have proved valuable in tackling a variety of tasks, ranging from concept categorization to relation extraction to many others (Sahlgren, 2006; Turney, 2006; Padó and Lapata, 2007). The typical approach in the field has been a "local" one, in which each semantic task (or set of closely related tasks) is treated as a separate problem, that requires its own corpus-derived model and algorithms. Its successes notwithstanding, the "one task – one model" approach has also some drawbacks.

From a cognitive angle, corpus-based models hold promise as simulations of how humans acquire and use conceptual and linguistic information from their environment (Landauer and Dumais, 1997). However, the common view in cognitive (neuro)science is that humans resort to a multipurpose *semantic memory*, i.e., a database of interconnected concepts and properties (Rogers and McClelland, 2004), adapting the information stored there to the task at hand. From an engineering perspective, going back to the corpus to train a different model for each application is inefficient and it runs the risk of overfitting the model to a specific task, while losing sight of its adaptivity – a highly desirable feature for any intelligent system. Think, by contrast, of WordNet, a single network of semantic information that has been adapted to all sorts of tasks, many of them certainly not envisaged by the resource creators.

In this paper, we explore a different approach to corpus-based semantics. Our model consists of a *distributional semantic memory* – a graph of weighted links between concepts - built once and for all from our source corpus. Starting from the tuples that can be extracted from this graph, we derive multiple semantic spaces to solve a wide range of tasks that exemplify various strands of corpus-based semantic research: measuring semantic similarity between concepts, concept categorization, selectional preferences, analogy of relations between concept pairs, finding pairs that instantiate a target relation and spotting an alternation in verb argument structure. Given a graph like the one in Figure 1 below, adaptation to all these tasks (and many others) can be reduced to two basic operations: 1) building semantic spaces, as co-occurrence matrices defined by choosing different units of the graph as row and column elements; 2) measuring similarity in the resulting matrix either between specific rows or between a row and an average of rows whose elements share a certain property.

After reviewing some of the most closely related work (Section 2), we introduce our approach (Section 3) and, in Section 4, we proceed to test it in various tasks, showing that its performance is always comparable to that of task-specific methods. Section 5 draws the current conclusions and discusses future directions.

2 Related work

Turney (2008) recently advocated the need for a uniform approach to corpus-based semantic tasks. Turney recasts a number of semantic challenges in terms of relational or analogical similarity. Thus, if an algorithm is able to tackle the latter, it can

Proceedings of the EACL 2009 Workshop on GEMS: GEometical Models of Natural Language Semantics, pages 1–8,
Athens, Greece, 31 March 2009.

also be used to address the former. Turney tests his system in a variety of tasks, obtaining good results across the board. His approach amounts to picking a task (analogy recognition) and reinterpreting other tasks as its particular instances. Conversely, we assume that each task may keep its specificity, and unification is achieved by designing a sufficiently general distributional structure, from which semantic spaces can be generated on demand. Currently, the only task we share with Turney is finding SAT analogies, where his method outperforms ours by a large margin (cf. Section 4.2.1). However, Turney uses a corpus that is 25 times larger than ours, and introduces negative training examples, whereas we dependency-parse our corpus – thus, performance is not directly comparable. Besides the fact that our approach does not require labeled training data like Turney's one, it provides, we believe, a more intuitive measure of taxonomic similarity (taxonomic neighbours are concepts that share similar contexts, rather than concepts that co-occur with patterns indicating a taxonomic relation), and it is better suited to model *productive* semantic phenomena, such as the selectional preferences of verbs with respect to unseen arguments (*eating topinambur* vs. *eating ideas*). Such tasks will require an extension of the current framework of Turney (2008) beyond evidence from the direct co-occurrence of target word pairs.

While our unified framework is, as far as we know, novel, the specific ways in which we tackle the different tasks are standard. Concept similarity is often measured by vectors of co-occurrence with context words that are typed with dependency information (Lin, 1998; Curran and Moens, 2002). Our approach to selectional preference is nearly identical to the one of Padó et al. (2007). We solve SAT analogies with a simplified version of the method of Turney (2006). Detecting whether a pair expresses a target relation by looking at shared connector patterns with model pairs is a common strategy in relation extraction (Pantel and Pennacchiotti, 2008). Finally, our method to detect verb slot similarity is analogous to the "slot overlap" of Joanis et al. (2008) and others. Since we aim at a unified approach, the lack of originality of our task-specific methods should be regarded as a positive fact: our general framework can naturally reproduce, locally, well-tried ad-hoc solutions.

3 Distributional semantic memory

Many different, apparently unrelated, semantic tasks resort to the same underlying information, a "distributional semantic memory" consisting of weighted *concept+link+concept* tuples extracted from the corpus. The *concepts* in the tuples are typically content words. The *link* contains corpus-derived information about how the two words are connected in context: it could be for example a dependency path or a shallow lexico-syntactic pattern. Finally, the *weight* typically derives from co-occurrence counts for the elements in a tuple, rescaled via entropy, mutual information or similar measures. The way in which the tuples are identified and weighted when populating the memory is, of course, of fundamental importance to the quality of the resulting models. However, once the memory has been populated, it can be used to tackle many different tasks, without ever having to go back to the source corpus.

Our approach can be compared with the typical organization of databases, in which multiple alternative "views" can be obtained from the same underlying data structure, to answer different information needs. The data structure is virtually independent from the way in which it is accessed. Similarly, the structure of our repository only obeys to the distributional constraints extracted from the corpus, and it is independent from the ways it will be "queried" to address a specific semantic task. Different tasks can simply be defined by how we split the tuples from the repository into row and column elements of a matrix whose cells are filled by the corresponding weights. Each of these derived matrices represents a particular *view* of distributional memory: we will discuss some of these views, and the tasks they are appropriate for, in Section 4.

Concretely, we used here the web-derived, 2-billion word ukWaC corpus,[1] dependency-parsed with MINIPAR.[2] Focusing for now on modeling noun-to-noun and noun-to-verb connections, we selected the 20,000 most frequent nouns and 5,000 most frequent verbs as target concepts (minus stop lists of very frequent items). We selected as target links the top 30 most frequent direct verb-noun dependency paths (e.g., *kill+obj+victim*), the top 30 preposition-mediated noun-to-noun or

[1] http://wacky.sslmit.unibo.it

[2] http://www.cs.ualberta.ca/~lindek/minipar.htm

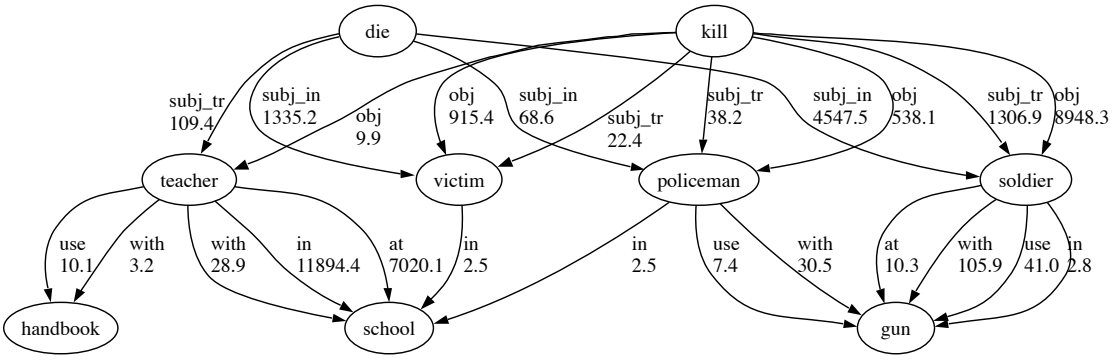


Figure 1: A fragment of distributional memory

verb-to-noun paths (e.g., *soldier+with+gun*) and the top 50 transitive-verb-mediated noun-to-noun paths (e.g., *soldier+use+gun*). We extracted all tuples in which a target link connected two target concepts. We computed the weight (strength of association) for all the tuples extracted in this way using the local MI measure (Evert, 2005), that is theoretically justified, easy to compute for triples and robust against overestimation of rare events. Tuples with local MI $\leq$ 0 were discarded. For each preserved tuple $c1+l+c2$, we added a same-weight $c1 + l^{-1} + c2$ tuple. In graph-theoretical terms (treating concepts as nodes and labeling the weighted edges with links), this means that, for each edge directed from $c1$ to $c2$, there is an edge from $c2$ to $c1$ with the same weight and inverse label, and that such inverse edges constitute the full set of links directed from $c2$ to $c1$. The resulting database (DM, for *Distributional Memory*) contains about 69 million tuples. Figure 1 depicts a fragment of DM represented as a graph (assume, for what we just said, that for each edge from x to y there is a same-weight edge from y to x with inverse label: e.g., the *obj* link from *kill* to *victim* stands for the tuples *kill+obj+victim* and *victim+obj*$^{-1}$*+kill*, both with weight 915.4; *subj_in* identifies the subjects of intransitive constructions, as in *The victim died*; *subj_tr* refers to the subjects of transitive sentences, as in *The policeman killed the victim*).

We also trained 3 closely comparable models that use the same source corpus, the same target concepts (in one case, also the same target links) and local MI as weighting method, with the same filtering threshold. The myPlain model implements a classic "flat" co-occurrence approach (Sahlgren, 2006) in which we keep track of verb-to-noun co-occurrence within a window that can include, maximally, one intervening noun, and noun-to-noun co-occurrence with no more than 2 intervening nouns. The myHAL model uses the same co-occurrence window, but, like HAL (Lund and Burgess, 1996), treats left and right co-occurrences as distinct features. Finally, myDV uses the same dependency-based target links of DM as filters. Like in the DV model of Padó and Lapata (2007), only pairs connected by target links are preserved, but the links themselves are not part of the model. Since none of these alternative models stores information about the links, they are only appropriate for the concept similarity tasks, where links are not necessary.

4 Semantic views and experiments

We now look at three views of the DM graph: *concept-by-link+concept* (CxLC), *concept+concept-by-link* (CCxL), and *concept+link-by-concept* (CLxC). Each view will be tested on one or more semantic tasks and compared with alternative models. There is a fourth possible view, *links-by-concept+concept* (LxCC), that is not explored here, but would lead to meaningful semantic tasks (finding links that express similar semantic relations).

4.1 The CxLC semantic space

Much work in computational linguistics and related fields relies on measuring similarity among words/concepts in terms of their patterns of co-occurrence with other words/concepts (Sahlgren, 2006). For this purpose, we arrange the information from the graph in a matrix where the concepts (nodes) of interest are rows, and the nodes they are connected to by outgoing edges are columns, typed with the corresponding edge label. We refer to this view as the *concept-by-link+concept*

(CxLC) semantic space. From the graph in Figure 1, we can for example construct the matrix in Table 1 (here and below, showing only some rows and columns of interest). By comparing the row vectors of such matrix using standard geometrical techniques (e.g., measuring the normalized cosine distance), we can find out about concepts that tend to share similar properties, i.e., are taxonomically similar (synonyms, antonyms, cohyponyms), e.g., soldiers and policemen, that both kill, are killed and use guns.

	subj_in^{-1} die	subj_tr^{-1} kill	obj^{-1} kill	with gun	use gun
teacher	109.4	0.0	9.9	0.0	0.0
victim	1335.2	22.4	915.4	0.0	0.0
soldier	4547.5	1306.9	8948.3	105.9	41.0
policeman	68.6	38.2	538.1	30.5	7.4

Table 1: A fragment of the CxLC space

We use the CxLC space in three taxonomic similarity tasks: modeling *semantic similarity judgments*, *noun categorization* and *verb selectional restrictions*.

4.1.1 Human similarity ratings

We use the dataset of Rubenstein and Goodenough (1965), consisting of 65 noun pairs rated by 51 subjects on a 0-4 similarity scale (e.g. *car-automobile* 3.9, *cord-smile* 0.0). The average rating for each pair is taken as an estimate of the perceived similarity between the two words. Following Padó and Lapata (2007), we use Pearson's r to evaluate how the distances (cosines) in the CxLC space between the nouns in each pair correlate with the ratings. Percentage correlations for DM, our other models and the best absolute result obtained by Padó and Lapata (DV+), as well as their best cosine-based performance (cosDV+), are reported in Table 2.

model	*r*	*model*	*r*
myDV	70	DV+	62
DM	**64**	myHAL	61
myPlain	63	cosDV+	47

Table 2: Correlation with similarity ratings

DM is the second-best model, outperformed only by DV when the latter is trained on comparable data (myDV in Table 2). Notice that, here and below, we did not try any parameter tuning (e.g., using a similarity measure different than cosine, feature selection, etc.) to improve the performance of DM.

4.1.2 Noun categorization

We use the concrete noun dataset of the ESSLLI 2008 Distributional Semantics shared task,[3] including 44 concrete nouns to be clustered into cognitively justified categories of increasing generality: 6-way (birds, ground animals, fruits, greens, tools and vehicles), 3-way (animals, plants and artifacts) and 2-way (natural and artificial entities). Following the task guidelines, we clustered the target row vectors in the CxLX matrix with CLUTO,[4] using its default settings, and evaluated the resulting clusters in terms of cluster-size-weighted averages of purity and entropy (see the CLUTO documentation). An ideal solution would have 100% purity and 0% entropy. Table 3 provides percentage results for our models as well as for the ESSLLI systems that reported all the relevant performance measures, indexed by first author. Models are ranked by a global score given by summing the 3 purity values and subtracting the 3 entropies.

model	*6-way*		*3-way*		*2-way*		*global*
	P	*E*	*P*	*E*	*P*	*E*	
Katrenko	89	13	100	0	80	59	197
Peirsman+	82	23	84	34	86	55	140
DM	**77**	**24**	**79**	**38**	**59**	**97**	**56**
myDV	80	28	75	51	61	95	42
myHAL	75	27	68	51	68	89	44
Peirsman–	73	28	71	54	61	96	27
myPlain	70	31	68	60	59	97	9
Shaoul	41	77	52	84	55	93	-106

Table 3: Concrete noun categorization

DM outperforms our models trained on comparable resources. Katrenko's system queries Google for patterns that cue the category of a concept, and thus its performance should rather be seen as an upper bound for distributional models. Peirsman and colleagues report results based on different parameter settings: DM's performance – not tuned to the task – is worse than their top model, but better than their worse.

4.1.3 Selectional restrictions

In this task we test the ability of the CxLC space to predict verbal selectional restrictions. We use the CxLC matrix to compare a concept to a "prototype" constructed by averaging a set of other concepts, that in this case represent typical fillers of

[3] http://wordspace.collocations.de/doku.php/esslli:start

[4] http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview

a verbal slot – for example, by averaging the vectors of the nouns that are, according to the underlying graph, objects of killing, we can build a vector for the typical "killee", and model selectional restrictions by measuring the similarity of other concepts (including concepts that have not been seen as objects of killing in the corpus) to this prototype. Note that the DM graph is used both to find the concepts to enter in the prototype (the set of nouns that are connected to a verb by the relevant edge) and to compute similarity. Thus, the method is fully unsupervised.

We test on the two datasets of human judgments about the plausibility of nouns as arguments (either subjects or objects) of verbs used in Padó et al. (2007), one (McRae) consisting of 100 noun-verb pairs rated by 36 subjects, the second (Padó) with 211 pairs rated by 20 subjects. For each verb in these datasets, we built its prototypical subject/object argument vector by summing the normalized vectors of the 50 nouns with the highest weight on the appropriate dependency link to the verb (e.g., the top 50 nouns connected to *kill* by an *obj* link). The cosine distance of a noun to a prototype is taken as the model "plausibility judgment" about the noun occurring as the relevant verb argument. Since we are interested in generalization, if the target noun is in the prototype set we subtract its vector from the prototype before calculating the cosine. For our comparison models, there is no way to determine which nouns would form the prototype, and thus we train them using the same top noun lists we employ for DM. Following Padó and colleagues, performance is measured by the Spearman ρ correlation coefficient between the average human ratings and the model predictions. Table 4 reports percentage coverage and correlations for our models as well as those in Padó et al. (2007) (ParCos is the best among their purely corpus-based systems).

model	McRae		Padó	
	coverage	ρ	*coverage*	ρ
Padó	56	41	97	51
DM	**96**	**28**	**98**	**50**
ParCos	91	21	98	48
myDV	96	21	98	39
myHAL	96	12	98	29
myPlain	96	12	98	27
Resnik	94	3	98	24

Table 4: Correlation with verb-argument plausibility judgments

DM does very well on this task: its performance on the Padó dataset is comparable to that of the Padó system, that relies on FrameNet. DM has nearly identical performance to the latter on the Padó dataset. On the McRae data, DM has a lower correlation, but much higher coverage. Since we are using a larger corpus than Padó et al. (2007), who train on the BNC, a fairer comparison might be the one with our alternative models, that are all outperformed by DM by a large margin.

4.2 The CCxL semantic space

Another view of the DM graph is exemplified in Table 5, where concept pairs are represented in terms of the edge labels (links) connecting them. Importantly, this matrix contains the same information that was used to build the CxLC space of Table 1, with a different arrangement of what goes in the rows and in the columns, but the same weights in the cells – compare, for example, the *soldier+gun*-by-*with* cell in Table 5 to the *soldier*-by-*with+gun* cell in Table 1.

		in	at	with	use
teacher	school	11894.4	7020.1	28.9	0.0
teacher	handbook	2.5	0.0	3.2	10.1
soldier	gun	2.8	10.3	105.9	41.0

Table 5: A fragment of the CCxL space

We use this space to measure "relational" similarity (Turney, 2006) of concept pairs, e.g., finding that the relation between teachers and handbooks is more similar to the one between soldiers and guns, than to the one between teachers and schools. We also extend relational similarity to prototypes. Given some example pairs instantiating a relation, we can harvest new pairs linked by the same relation by computing the average CCxL vector of the examples, and finding the nearest neighbours to this average. In the case at hand, the link profile of pairs such as *soldier+gun* and *teacher+handbook* could be used to build an "instrument relation" prototype.

We test the CCxL semantic space on *recognizing SAT analogies* (relational similarity between pairs) and *semantic relation classification* (relational similarity to prototypes).

4.2.1 Recognizing SAT analogies

We used the set of 374 multiple-choice questions from the SAT college entrance exam. Each question includes one target pair, usually called

the stem (*ostrich-bird*) , and 5 other pairs (*lion-cat*, *goose-flock*, *ewe-sheep*, *cub-bear*, *primate-monkey*). The task is to choose the pair most analogous to the stem. Each SAT pair can be represented by the corresponding row vector in the CCxL matrix, and we select the pair with the highest cosine to the stem. In Table 6 we report our results, together with the state-of-the-art from the ACL wiki[5] and the scores of Turney (2008) (PairClass) and from Amaç Herdağdelen's PairSpace system, that was trained on ukWaC. The Attr cells summarize the performance of the 6 models on the wiki table that are based on "attributional similarity" only (Turney, 2006). For the other systems, see the references on the wiki. Since our coverage is very low (44% of the stems), in order to make a meaningful comparison with the other models, we calculated a corrected score (DM−). Having full access to the results of the ukWaC-trained, similarly performing PairSpace system, we calculated the adjusted score by assuming that the DM-to-PairSpace error ratio (estimated on the items we cover) is constant on the whole dataset, and thus the DM hit count on the unseen items is approximated by multiplying the PairSpace hit count on the same items by the error ratio (DM+ is DM's accuracy on the covered test items only).

model	*% correct*	*model*	*% correct*
LRA	56.1	KnowBest	43.0
PERT	53.3	**DM−**	**42.3**
PairClass	52.1	LSA	42.0
VSM	47.1	AttrMax	35.0
DM+	**45.3**	AttrAvg	31.0
PairSpace	44.9	AttrMin	27.3
k-means	44.0	Random	20.0

Table 6: Accuracy with SAT analogies

DM does not excel in this task, but its corrected performance is well above chance and that of all the attributional models, and comparable to that of a WordNet-based system (KnowBest) and a system that uses manually crafted information about analogy domains (LSA). All systems with performance above DM+ (and *k*-means) use corpora that are orders of magnitude larger than ukWaC.

4.2.2 Classifying semantic relations

We also tested the CCxL space on the 7 semantic relations between nominals adopted in Task 4 of SEMEVAL 2007 (Girju et al., 2007): Cause-Effect, Instrument-Agency, Product-Producer, Origin-Entity, Theme-Tool, Part-Whole, Content-Container. For each relation, the dataset includes 140 training examples and about 80 test cases. Each example consists of a small context retrieved from the Web, containing word pairs connected by a certain pattern (e..g., "* contains *"). The retrieved contexts were manually classified by the SEMEVAL organizers as positive (e.g., *wrist-arm*) or negative (e.g., *effectiveness-magnesium*) instances of a certain relation (e.g., Part-Whole). About 50% training and test cases are positive instances. For each relation, we built "hit" and "miss" prototype vectors, by averaging across the vectors of the positive and negative training pairs attested in our CCxL model (we use only the word pairs, not the surrounding contexts). A test pair is classified as a hit for a certain relation if it is closer to the hit prototype vector for that relation than to the corresponding miss prototype. We used the SEMEVAL 2007 evaluation method, i.e., precision, recall, F-measure and accuracy, macroaveraged over all relations, as reported in Table 7. The DM+ scores ignore the 32% pairs not in our CCxL space; the DM− scores assume random performance on such pairs. These scores give the range within which our performance will lie once we introduce techniques to deal with unseen pairs. We also report results of the SEMEVAL systems that did not use the organizer-provided WordNet sense labels nor information about the query used to retrieve the examples, as well as performance of several trivial classifiers, also from the SEMEVAL task description.

model	*precision*	*recall*	*F*	*accuracy*
UCD-FC	66.1	66.7	64.8	66.0
UCB	62.7	63.0	62.7	65.4
ILK	60.5	69.5	63.8	63.5
DM+	**60.3**	**62.6**	**61.1**	**63.3**
UMELB-B	61.5	55.7	57.8	62.7
SemeEval avg	59.2	58.7	58.0	61.1
DM−	**56.7**	**58.2**	**57.1**	**59.0**
UTH	56.1	57.1	55.9	58.8
majority	81.3	42.9	30.8	57.0
probmatch	48.5	48.5	48.5	51.7
UC3M	48.2	40.3	43.1	49.9
alltrue	48.5	100.0	64.8	48.5

Table 7: SEMEVAL relation classification

The DM accuracy is higher than the three SEMEVAL baselines (majority, probmatch and alltrue), DM+ is above the average performance of

[5] http://www.aclweb.org/aclwiki/index.php?title=SAT_Analogy_Questions

the comparable SEMEVAL models. Differently from DM, the models that outperform it use features extracted from the training contexts and/or specific additional resources: an annotated compound database for UCD-FC, machine learning algorithms to train the relation classifiers (ILK, UCD-FC), Web counts (UCB), etc. The less than optimal performance by DM is thus counterbalanced by its higher "parsimony" and generality.

4.3 The CLxC semantic space

A third view of the information in the DM graph is the *concept+link-by-concept* (CLxC) semantic space exemplified by the matrix in Table 8.

		teacher	victim	soldier	policeman
kill	subj_tr	0.0	22.4	1306.9	38.2
kill	obj	9.9	915.4	8948.3	538.1
die	subj_in	109.4	1335.2	4547.5	68.6

Table 8: A fragment of the CLxC space

This view captures patterns of similarity between (surface approximations to) argument slots of predicative words. We can thus use the CLxC space to extract generalizations about the inner structure of lexico-semantic representations of the sort formal semanticists have traditionally being interested in. In the example, the patterns of co-occurrence suggest that objects of killing are rather similar to subjects of dying, hinting at the classic *cause(subj,die(obj))* analysis of killing by Dowty (1977) and many others. Again, no new information has been introduced – the matrix in Table 8 is yet another re-organization of the data in our graph (compare, for example, the *die+subj_in*-by-*teacher* cell of this matrix with the *teacher*-by-*subj_in+die* cell in Table 1).

4.3.1 The causative/inchoative alternation

Syntactic alterations (Levin, 1993) represent a key aspect of the complex constraints that shape the syntax-semantics interface. One of the most important cases of alternation is the *causative/inchoative*, in which the object argument (e.g., *John broke the vase*) can also be realized as an intransitive subject (e.g., *The vase broke*). Verbs differ with respect to the possible syntactic alternations they can participate in, and this variation is strongly dependent on their semantic properties (e.g. semantic roles, event type, etc.). For instance, while *break* can undergo the causative/inchoative alternation, *mince* cannot: cf. *John minced the meat* and **The meat minced*.

We test our CLxC semantic space on the discrimination between transitive verbs undergoing the causative-inchoative alternations and non-alternating ones. We took 232 causative/inchoative verbs and 170 non-alternating transitive verbs from Levin (1993). For each verb v_i, we extracted from the CLxC matrix the row vectors corresponding to its transitive subject ($v_i + subj\_tr$), intransitive subject ($v_i + subj\_in$), and direct object ($v_i + obj$) slots. Given the definition of the causative/inchoative alternation, we predict that with alternating verbs $v_i + subj\_in$ should be similar to $v_i + obj$ (the things that are broken also break), while this should not hold for non-alternating verbs (mincees are very different from mincers).

Our model is completely successful in detecting the distinction. The cosine similarity between transitive subject and object slots is fairly low for both classes, as one would expect (medians of 0.16 for alternating verbs and 0.11 for non-alternating verbs). On the other hand, while for the non-alternating verbs the median cosine similarity between the intransitive subject and object slots is a similarly low 0.09, for the alternating verbs the median similarity between these slots jump up to 0.31. Paired t-tests confirm that the per-verb difference between transitive subject vs. object cosines and intransitive subject vs. object cosines is highly statistically significant for the alternating verbs, but not for the non-alternating ones.

5 Conclusion

We proposed an approach to semantic tasks where statistics are collected only once from the source corpus and stored as a set of weighted *concept+link+concept* tuples (naturally represented as a graph). Different semantic spaces are constructed on demand from this underlying "distributional memory", to tackle different tasks without going back to the corpus. We have shown that a straightforward implementation of this approach leads to excellent performance in various taxonomic similarity tasks, and to performance that, while not outstanding, is at least reasonable on relational similarity. We also obtained good results in a task (detecting the causative/inchoative alternation) that goes beyond classic NLP applications and more in the direction of theoretical semantics.

The most pressing issue we plan to address is how to improve performance in the relational sim-

ilarity tasks. Fortunately, some shortcomings of our current model are obvious and easy to fix. The low coverage is in part due to the fact that our set of target concepts does not contain, by design, some words present in the task sets. Moreover, while our framework does not allow ad-hoc optimization of corpus-collection methods for different tasks, the way in which the information in the memory graph is adapted to tasks should of course go beyond the nearly baseline approaches we adopted here. In particular, we need to develop a backoff strategy for unseen pairs in the relational similarity tasks, that, following Turney (2006), could be based on constructing surrogate pairs of taxonomically similar words found in the CxLC space.

Other tasks should also be explored. Here, we viewed our distributional memory in line with how cognitive scientists look at the semantic memory of healthy adults, i.e., as an essentially stable long term knowledge repository. However, much interesting semantic action takes place when underlying knowledge is adapted to context. We plan to explore how contextual effects can be modeled in our framework, focusing in particular on how composition affects word meaning (Erk and Padó, 2008). Similarity could be measured directly on the underlying graph, by relying on graph-based similarity algorithms – an elegant approach that would lead us to an even more unitary view of what distributional semantic memory is and what it does. Alternatively, DM could be represented as a three-mode tensor in the framework of Turney (2007), enabling smoothing operations analogous to singular value decomposition.

Acknowledgments

We thank Ken McRae and Peter Turney for providing data-sets, Amaç Herdağdelen for access to his results, Katrin Erk for making us look at DM as a graph, and the reviewers for helpful comments.

References

J. Curran and M. Moens. 2002. Improvements in automatic thesaurus extraction. *Proceedings of the ACL Workshop on Unsupervised Lexical Acquisition*, 59–66.

D. Dowty. 1977. *Word meaning and Montague Grammar*. Kluwer, Dordrecht.

K. Erk and S. Padó. 2008. A structured vector space model for word meaning in context. *Proceedings of EMNLP 2008*.

S. Evert. 2005. *The statistics of word cooccurrences*. Ph.D. dissertation, Stuttgart University, Stuttgart.

R. Girju, P. Nakov, V. Nastase, S. Szpakowicz, P. Turney and Y. Deniz. 2007. SemEval-2007 task 04: Classification of semantic relations between nominals. *Proceedings of SemEval-2007*, 13–18.

E. Joanis, S. Stevenson and D. James. 2008. A general feature space for automatic verb classification. *Natural Language Engineering*, 14(3): 337–367.

T.K. Landauer and S.T. Dumais. 1997. A solution to Plato's problem: The Latent Semantic Analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104(2): 211–240.

B. Levin. 1993. *English Verb Classes and Alternations. A Preliminary Investigation*. Chicago, University of Chicago Press.

D. Lin. 1998. Automatic retrieval and clustering of similar words. *Proceedings of ACL 1998*, 768–774.

K. Lund and C. Burgess. 1996. Producing high-dimensional semantic spaces from lexical cooccurrence. *Behaviour Research Methods*, 28: 203–208.

S. Padó and M. Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2): 161–199.

S. Padó, S. Padó and K. Erk. 2007. Flexible, corpus-based modelling of human plausibility judgements. *Proceedings EMNLP 2007*, 400–409.

P. Pantel and M. Pennacchiotti. 2008. Automatically harvesting and ontologizing semantic relations. In P. Buitelaar and Ph. Cimiano (eds.), *Ontology learning and population*. IOS Press, Amsterdam.

T. Rogers and J. McClelland. 2004. *Semantic cognition: A parallel distributed processing approach*. The MIT Press, Cambridge.

H. Rubenstein and J.B. Goodenough. 1965. "Contextual correlates of synonymy". *Communications of the ACM*, 8(10):627-633.

M. Sahlgren. 2006. *The Word-space model*. Ph.D. dissertation, Stockholm University, Stockholm.

P. Turney. 2006. Similarity of semantic relations. *Computational Linguistics*, 32(3): 379–416.

P. Turney. 2007. Empirical evaluation of four tensor decomposition algorithms. *IIT Technical Report ERB-1152*, National Research Council of Canada, Ottawa.

P. Turney. 2008. A uniform approach to analogies, synonyms, antonyms and associations. *Proceedings of COLING 2008*, 905–912.

Word Space Models of Lexical Variation

Yves Peirsman
Research Foundation – Flanders &
QLVL, University of Leuven
Leuven, Belgium
yves.peirsman@arts.kuleuven.be

Dirk Speelman
QLVL, University of Leuven
Leuven, Belgium
dirk.speelman@arts.kuleuven.be

Abstract

In the recognition of words that are typical of a specific language variety, the classic keyword approach performs rather poorly. We show how this keyword analysis can be complemented with a word space model constructed on the basis of two corpora: one representative of the language variety under investigation, and a reference corpus. This combined approach is able to recognize the markers of a language variety as words that not only have a significantly higher frequency as compared to the reference corpus, but also a different distribution. The application of word space models moreover makes it possible to automatically discover the lexical alternative to a specific marker in the reference corpus.

1 Introduction

Different varieties of the same language often come with their lexical peculiarities. Some words may be restricted to a specific register, while other ones may have different meanings in different regions. In corpus linguistics, the most straightforward way of finding such words that are typical of one language variety is to compile a corpus of that variety and compare it to a reference corpus of another variety. The most obvious comparison takes on the form of a keyword analysis, which looks for the words that are significantly more frequent in the one corpus as compared to the other (Dunning, 1993; Scott, 1997; Rayson et al., 2004). For the purposes of a language-variational study, this classic keyword approach often does not suffice, however. As Kilgarriff has argued, keyword statistics are far too sensitive to high frequencies or topical differences to be used in the study of vocabulary differences (Kilgarriff, 2001). We therefore put forward an approach that combines keyword statistics with distributional models of lexical semantics, or word space models (Sahlgren, 2006; Bullinaria and Levy, 2007; Padó and Lapata, 2007; Peirsman, 2008). In this way, we not only check whether two words have significantly different frequencies in the two relevant language varieties, but also to what degree their distribution varies between the corpora.

In this paper, we will focus on the lexical differences between two regional varieties of Dutch. Dutch is interesting because it is the official language of two neighbouring countries, Belgium and the Netherlands. Between these two countries, there exists a considerable amount of lexical variation (Speelman et al., 2006). There are words much more frequently used in one of the two varieties as well as terms that have a different meaning in the two regions. We will call such words *markers* of a specific *lect* — a general term for regiolects, dialects, or other language varieties that are specific to a certain region, genre, etc. By constructing a word space model on the basis of two corpora instead of one, we will show how the distributional approach to lexical semantics can aid the recognition of such lectal variation.

In the next section, we will point out the weaknesses of the classic keyword approach, and show how word space models can provide a solution. In section 3, we will discuss how our approach recognizes markers of a given lect. In section 4, we will demonstrate how it can automatically find the alternatives in the other language variety. Section 5 wraps up with conclusions and an outlook for future research.

2 Bilectal Word Spaces

Intuitively, the most obvious way of looking for words that mark a particular language variety, is to take a corpus that represents this variety, and calculate its keywords with respect to a reference

Proceedings of the EACL 2009 Workshop on GEMS: GEometical Models of Natural Language Semantics, pages 9–16,
Athens, Greece, 31 March 2009.

χ^2		log-likelihood	
keyword	χ^2	keyword	log-likelihood
frank/noun (‘franc’)	262492.0	frank/noun (‘franc’)	335587.3
meer/adj (‘more’)	149505.0	meer/adj (‘more’)	153811.6
foto/noun (‘photograph’)	84286.7	Vlaams/adj (‘Flemish’)	93723.2
Vlaams/adj (‘Flemish’)	83663.0	foto/noun (‘photograph’)	87235.1
veel/adj (‘much’/’many’)	73655.5	vrijdag/noun (‘Friday’)	77865.5
Belgisch/adj (‘Belgian’)	62280.2	veel/adj (‘much’/‘many’)	74167.1
vrijdag/noun (‘Friday’)	59135.9	Belgisch/adj (‘Belgian’)	64786.0
toekomst/noun (‘future’)	42440.5	toekomst/noun (‘future’)	55879.1
dossier/noun (‘file’)	34623.3	dossier/noun (‘file’)	45570.0
Antwerps/adj (‘Antwerp’)	33659.1	ziekenhuis/noun (’hospital’)	44093.3

Table 1: Top 10 keywords for the Belgian newspaper corpus, as compared to the Twente Nieuws Corpus.

corpus (Dunning, 1993; Scott, 1997; Rayson et al., 2004). This keyword approach has two important weaknesses, however. First, it has been shown that statistically significant differences in the relative frequencies of a word may arise from high absolute frequencies rather than real lexical variation (Kilgarriff, 2001). Second, in the explicit comparison of two language varieties, the keyword approach offers no way of telling what word in the reference corpus, if any, serves as the alternative to an identified marker. Word space models offer a solution to both of these problems.

We will present this solution on the basis of two corpora of Dutch. The first is the Twente Nieuws Corpus (TwNC), a 300 million word corpus of Netherlandic Dutch newspaper articles from between 1999 and 2002. The second is a corpus of Belgian Dutch we compiled ourselves, with the goal of making it as comparable to the Twente Nieuws Corpus as possible. With newspaper articles from six major Belgian newspapers from the years 1999 to 2005, it totals over 1 billion word tokens. Here we will work with a subset of this corpus of around 200 million word tokens.

2.1 Keywords

As our starting point, we calculated the keywords of the Belgian corpus with respect to the Netherlandic corpus, both on the basis of a chi-square test (with Yates’ continuity correction) (Scott, 1997) and the log-likelihood ratio (Dunning, 1993). We considered only words with a total frequency of at least 200 that moreover occurred at least five times in each of the five newspapers that make up the Belgian corpus. This last restriction was imposed in order to exclude idiosyncratic language use in any of those newspapers. The top ten resulting keywords, listed in Table 1, show an overlap of 90% between the tests. The words fall into a number of distinct groups. *Frank*, *Vlaams*, *Belgisch* and *Antwerps* (this last word appears only in the χ^2 top ten) indeed typically occur in Belgian Dutch, simply because they are so tightly connected with Belgian culture. *Dossier* may reflect a Belgian preference for this French loanword. Why the words *meer*, *veel*, *vrijdag*, *toekomst* and *ziekenhuis* (only in the log-likelihood top ten) are in the lists, however, is harder to explain. There does not appear to be a linguistically significant difference in use between the two language varieties, neither in frequency nor in sense. The presence of *foto*, finally, may reflect certain publishing habits of Belgian newspapers, but again, there is no obvious difference in use between Belgium and the Netherlands. In sum, these Belgian keywords illustrate the weakness of this approach in the modelling of lexical differences between two language varieties. This problem was already noted by Kilgarriff (2001), who argues that “[t]he LOB-Brown differences cannot in general be interpreted as British-American differences”. One of the reasons is that “for very common words, high χ^2 values are associated with the sheer quantity of evidence and are not necessarily associated with a pre-theoretical notion of distinctiveness”. One way to solve this issue is presented by Speelman et al. (2008). In their so-called *stable lexical markers* analysis, the word frequencies in one corpus are compared to those in several reference corpora. The keyness of a word then corresponds to the number of times it appears in the resulting keyword lists of the first corpus. This repetitive test

helps filter out spurious keywords whose statistical significance does not reflect a linguistically significant difference in frequency. Here we explore an alternative solution, which scores candidate markers on the basis of their contextual distribution in the two corpora, in a so-called bilectal word space.

2.2 Bilectal Word Spaces

Word space models (Sahlgren, 2006; Bullinaria and Levy, 2007; Padó and Lapata, 2007; Peirsman, 2008) capture the semantic similarity between two words on the basis of their distribution in a corpus. In these models, two words are similar when they often occur with the same context words, or when they tend to appear in the same syntactic relationships. For our purposes, we need to build a word space on the basis of two corpora, more or less in the vein of Rapp's (1999) method for the identification of translation equivalents. The main difference is that we use two corpora of the same language, each of which should be representative of one of the language varieties under investigation. All other variables should be kept as constant as possible, so that we can attribute differences in word use between the two corpora to lexical differences between the two lects. Next, we select the words that occur in both corpora (or a subset of the n most frequent words to reduce dimensionality) as the dimensions of the word space model. For each target word, we then build two context vectors, one for each corpus. These context vectors contain information about the distribution of the target word. We finally calculate the similarity between two context vectors as the cosine of the angle between them.

One crucial parameter in the construction of word space models is their definition of *distribution*. Some models consider the syntactic relationships in which a target word takes part (Padó and Lapata, 2007), while other approaches look at the collocation strength between a target and all of the words that occur within n words to its left and right (Bullinaria and Levy, 2007). With these last *word-based* approaches, it has been shown that small context sizes in particular lead to good models of the semantic similarity between two words (Bullinaria and Levy, 2007; Peirsman, 2008). So far, we have therefore performed experiments with context sizes of one, two and three words to the left and right of the target. These all gave very similar results. Experiments with other context sizes and with syntactic features will be carried out in the near future. In this paper, we report on the results of a word-based model with context size three.

In order to identify the markers of Belgian Dutch, we start from the keyword lists above. For each of the keywords, we get their context vector from the Belgian corpus, and find the 100 most similar context vectors from the Netherlandic corpus. The words that correspond to these context vectors are called the 'nearest neighbours' to the keyword. In the construction of our word space model, we selected from both corpora the 4,000 most frequent words, and used the cross-section of 2,538 words as our set of dimensions or context features. The model then calculated the point-wise mutual information between the target and each of the 2,538 context words that occurred at least twice in its context. All words in the Netherlandic Dutch corpus with a frequency of at least 200, plus the target itself, were considered possible nearest neighbours to the target.

Generally, where there are no major differences in the use of a keyword between the two lects, it will have itself as its nearest neighbour. If this is not the case, this may identify the keyword as a marker of Belgian Dutch. For example, six words from the lists above have themselves as their nearest neighbour: *meer*, *foto*, *veel*, *vrijdag*, *toekomst* and *ziekenhuis*. These are indeed the keywords that made little sense from a language-variational perspective. *Dossier* is its own second nearest neighbour, which indicates that there is slightly less of a match between its Belgian and Netherlandic use. Finally, the words linked to Belgian culture — *frank*, *Vlaams*, *Belgisch* and *Antwerps* — are much lower in their own lists of nearest neighbours, or totally absent, which correctly identifies them as markers of Belgian Dutch. In short, the keyword analysis ensures that the word occurs much more frequently in Belgian Dutch than in Netherlandic Dutch; the word space approach checks if it also has a different distribution in the two corpora.

For markers of Belgian Dutch, we can interpret the nearest neighbour suggested by the system as the other variety's alternative to that marker. For instance, *dossier* has *rapport* as its nearest neighbour, a synonym which indeed has a high keyword value for our Netherlandic Dutch corpus. Similarly, the culture-related words have their Dutch

equivalents as their distributionally most similar words: *frank* has *gulden* ('guilder'), *Vlaams* and *Belgisch* both have *Nederlands* ('Dutch'), and *Antwerps* has *Amsterdams* ('Amsterdam (adj.)'). This makes intuitive sense if we take meaning to be a relative concept, where for instance a concept like 'currency of this country' is instantiated by the franc in Belgium and the guilder in Holland — at least in the pre-Euro period. These findings suggest that our combined method can be applied more generally in order to automatically discover lexical differences between the two language varieties.

3 Recognizing lectal differences

First we want to investigate whether a bilectal word space model can indeed contribute to the correct identification of markers of Belgian Dutch on a larger scale. We therefore had both types of approaches — the simple keyword approach and the combined method — suggest a top 2,000 of possible markers on the basis of our two corpora. The combined approach uses the same word space method we described above, with 2,538 dimensions and a context size of three. Basing itself on the lists of nearest neighbours, it then reorders the list of keywords, so as to arrive at a ranking that reflects lectal variation better than the original one. To this goal, each keyword receives a new score, which is the multiplication of two individual numbers. The first number is its rank in the original keyword list. At this point we considered only the 5,000 highest scoring keywords. The second is based on a list that ranks the words according to their difference in distribution between the two corpora. Words that do not occur in their own list of 100 nearest neighbours appear at the top of the list (rank 1), followed by words that are their own 100th nearest neighbour (rank 2), and so on to the words that have themselves as nearest neighbour (rank 101). In the future we plan to consider different numbers of neighbours in order to punish words with very different distributions more or less heavily. At this stage, however, restricting the method to 100 nearest neighbours gives fine results. These two ranks are then multiplied to give a combined score, on the basis of which a final list of candidates for lectal variation is computed. The lower this combined score (reflecting either high keyword values, very different distributions in the two corpora, or both), the higher the likelihood that the word is a marker of Belgian Dutch. This approach thus ensures that words that have very different distributions in the two corpora are promoted with respect to the original keyword list, while words with very similar distributions are downgraded.

candidate marker	evaluation
frank/noun ('franc')	culture
Vlaams/adj ('Flemish')	culture
match/noun ('match')	literature
info/noun ('info')	
rijkswacht/noun ('state police')	RBBN
weekend/noun ('weekend')	
schepen/noun ('alderman')	RBBN
fr./noun ('franc')	culture
provinciaal/adj ('provincial')	RBBN
job/noun ('job')	RBBN

Table 2: Top ten candidate markers suggested by the combined method on the basis of the log-likelihood ratio.

As our Gold Standard we used the *Reference List of Belgian Dutch* (*Referentiebestand Belgisch Nederlands*, RBBN), a list of almost 4,000 words and expressions that are typical of Belgian Dutch (Martin, 2005). These are classified into a number of groups — culturally-related terms (e.g., names of political parties), Belgian markers that are not lexicalized in Netherlandic Dutch, markers that are lexicalized in Netherlandic Dutch, etc. We used a subset of 717 one-word nouns, verbs and adjectives that appear at least 200 times in our Belgian corpus to evaluate our approach. Even if we informally explore the first ten candidate markers, the advantages of combining the log-likelihood ratio with the word space model already become clear (see table 2). Four of these candidates are in the RBBN gold standard. Similarly, *frank*, *Vlaams* and *fr.* are culturally related to Belgium, while *match* has been identified as a typically Belgian word in previous corpus-linguistic research (Geeraerts et al., 1999). *Info* and *weekend* are not present in the external sources we consulted, but nevertheless show an interesting distribution with respect to their respective synonyms. In the Belgian corpus, *info* occurs more often than the longer and more formal *information* (32,009 vs 30,171), whereas in the Dutch corpus the latter is used about 25 times as frequently as the former (1,681 vs 41,429). Similarly, the Belgian corpus

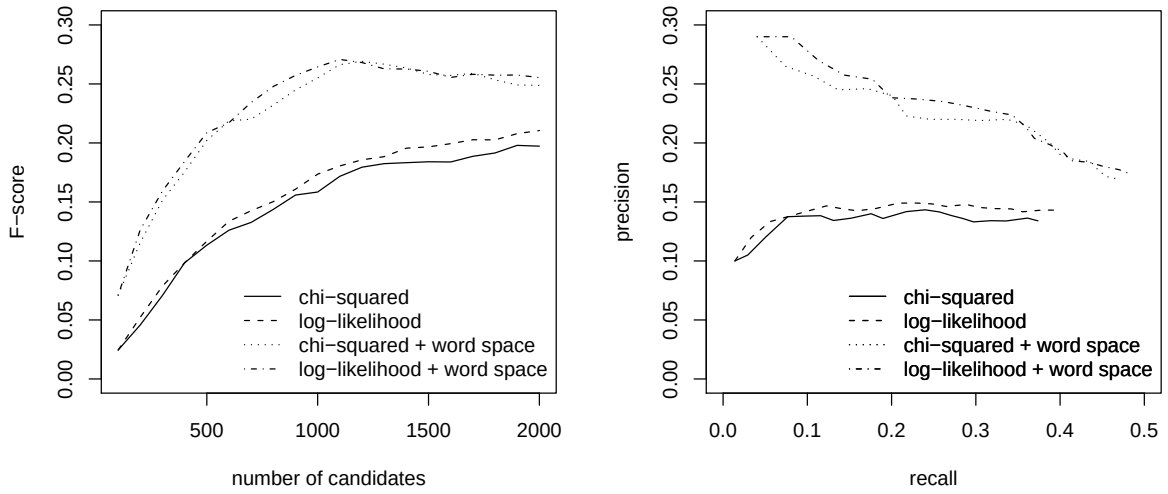


Figure 1: Precision and recall figures of the keyword methods and the combined approaches.

contains far more instances of *weekend* than of its synonym *weekeinde* (35,406 vs 6,390), while the Dutch corpus shows the reverse pattern (6,974 vs 28,234). These words are thus far better candidate markers than the original keywords *meer*, *foto*, *veel*, *vrijdag*, *toekomst* or *ziekenhuis*, which have disappeared from the top ten.

Let us now evaluate the methods more broadly, on the basis of the top 2,000 keywords they suggest. The left plot in Figure 1 shows their F-scores in function of the number of suggested markers; the right graph plots precision in function of recall. The two keyword approaches score rather similarly, with the log-likelihood ratio achieving slightly better results than the chi-square test. This superiority of the log-likelihood approach was already noted by Rayson et al. (2004). Both combined methods give a very clear advantage over the simple keyword statistics, again with the best results for the log-likelihood ratio. For example, ten of the first 100 candidates suggested by both keyword approaches are present in our Gold Standard, giving a precision of 10% and a recall of 1.4% (F-score 2.4%). Adding our word space model makes this figure rise to 29 correct markers, resulting in a precision of 29% and a recall of 4% (F-score 7.1%). This large advantage in performance is maintained further down the list. At 1000 candidates, the keyword approaches have a recall of around 20% (chi-square 19%, log-likelihood 21%) and a precision of around 14% (chi-square 14%, log-likelihood 15%). At the same point, the combined approaches have reached a recall of over 30% (chi-square 31%, log-likelihood 32%) with a precision of around 22% (chi-square 22%, log-likelihood 23%). Expressed differently, the best keyword approach needs around 500 candidates to recover 10% of the gold standard, 1000 to recover 20% and 2000 to recover 40%. This linear increase is outperformed by the best combined approach, which needs only 300, 600 and 1500 candidate markers to reach the same recall figures. This corresponds to relative gains of 40%, 40% and 25%. As these results indicate, the performance gain starts to diminish after 1000 candidates. Future experiments will help determine if this issue can be resolved with different parameter settings.

Despite these large gains in performance, the combined method still has problems with a number of Belgian markers. A manual analysis of these cases shows that they often have several senses, only one of which is typical of Belgian Dutch. The Reference List for instance contains *fout* ('mistake') and *mossel* ('mussel') as Belgian markers, with their specialized meanings 'foul (in sports)' and 'weakling'. Not only do these words have very low keyword values for the Belgian corpus; they also have very similar distributions in the two corpora, and are their own first and second neighbour, respectively. Sometimes a failure to recognize a particular marker is more due

	top 100		top 500	
class	n	%	n	%
in Gold Standard	29	29%	127	25.4%
in Van Dale	11	22%	47	9.4%
related	2	2%	23	4.6%
cultural terms	25	25%	60	12%
total	67	67%	257	51.4%

Table 3: Manual analysis of the top 500 words suggested by the combined approach.

to the results of one individual method. This is for instance the case with the correct Belgian marker *home* ('(old people's) home'). Although the word space model does not find this word in its own list of nearest Netherlandic neighbours, it remains low on the marker list due to its fairly small log-likelihood ratio. Conversely, *punt*, *graad* and *klaar* are rather high on the keyword list of the Belgian corpus, but are downgraded, as they have themselves as their nearest neighbour. This is again because their status as a marker only applies to one infrequent meaning ('school mark', 'two-year cycle of primary education' and 'clear') instead of the dominant meanings ('final stop, point (e.g., in sports)', 'degree' and 'ready'), which are shared between the two regional varieties. However, this last disadvantage applies to all markers that are much more frequently used in Belgium but still sometimes occur in the Netherlandic corpus with a similar distribution.

Finally, because our Gold Standard is not an exhaustive list of Belgian Dutch markers, the results in Figure 1 are an underestimate of real performance. We therefore manually went through the top 500 markers suggested by the best combined approach and classified them into three new groups. The results of this analysis are presented in Table 3. First, we consulted the *Van Dale Groot Woordenboek der Nederlandse taal* (Den Boon and Geeraerts, 2005), the major dictionary of Dutch, which contains about 3,000 words marked with the label "Belgian Dutch". 11% of the first 100 and 9.4% of the first 500 candidates that were initially judged incorrect carry this label or have a definition that explicitly refers to Belgium. Second, we counted the words that are morphologically related to words in the Gold Standard or to Belgian words found in Van Dale. These are for instance compound nouns one of whose parts is present in the Gold Standard, which means that

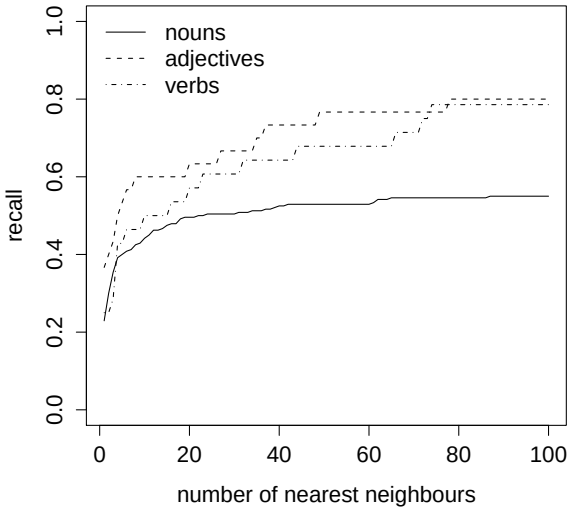


Figure 2: Percentage of markers of Belgian Dutch whose Netherlandic alternative is present among their n nearest neighbours.

they are correct markers of Belgian Dutch as well. They represent 2% of the top 100 and 4.6% of the top 500. Third, we counted the words that are inherently linked to Belgian culture, mostly in the form of place names. This group corresponds to 25% of the first 100 and 12% of the first 500 candidate markers. This suggests that the true precision of our method at 100 and 500 candidates is thus at least 67% and 51.4%, respectively.

4 Finding alternatives

The *Reference List of Belgian Dutch* not only lists Belgian Dutch words and expressions, but also gives their Netherlandic Dutch alternative, if one exists. Our word space model offers us a promising way of determining this alternative automatically, by looking at the nearest Netherlandic neighbours to a Belgian marker. As our Gold Standard, we selected from the Reference List those words with a frequency of at least 200 in the Belgian corpus whose Dutch alternative also had a frequency of at least 200 in the Dutch corpus. This resulted in a test set of 315 words: 240 nouns, 45 verbs and 30 adjectives. For each of these words, we used our word space model to find the 100 nearest Netherlandic neighbours, again with context size three but now with as dimensions all words shared between the two corpora, in order to improve performance. We then determined if the

Dutch alternative was indeed in the list of nearest neighbours to the target. We started by looking at the single nearest neighbour only, and then step by step extended the list to include the 100 nearest neighbours. If a word had itself among its nearest neighbours, this neighbour was discarded and replaced by the next one down the list. The results are shown in Figure 2. 11 out of 30 adjectives (36.7%), 10 out of 45 verbs (22.2%) and 56 out of 240 nouns (23.3%) had their Dutch alternative as their nearest neighbour. At ten nearest neighbours, these figures had risen to 60.0%, 48.9% and 44.6%. These encouraging results underpin the usefulness of word space models in language-variational research.

A manual analysis of Belgian markers for which the approach does not find the Netherlandic alternative again reveals that a large majority of these errors occur when polysemous words have only one, infrequent meaning that is typical of Belgian Dutch. For example, the dominant sense of the word *tenor* is obviously the 'male singer' meaning. In Belgium, however, this term can also refer to a leading figure, for instance in a political party or a sports discipline. Since this metaphorical sense is far less frequent than the literal one, the context vector fails to pick it up, and almost all nearest Netherlandic neighbours are related to opera or music. One way to solve this problem would be to abandon word space models that build only one context vector per word. Instead, we could cluster all individual contexts of a word, with the aim of identifying context clusters that correspond to the several senses of that word (Schütze, 1998). This is outside the scope of the current paper, however.

5 Conclusions and future research

We have presented an application of word space models to language-variational research. To our knowledge, the construction of word space models on the basis of two corpora of the same language instead of one is new to both variational linguistics and Natural Language Processing. It complements the classic keyword approach in that it helps recognize those keywords that, in addition to their different relative frequencies in two language varieties, also have a substantially different distribution. An application of this method to Belgian Dutch showed that the keywords that pass this test indeed much more often represent markers of the language variety under investigation. Moreover, often the word space model also succeeded in identifying the Netherlandic Dutch alternative to the Belgian marker.

As the development of this approach is still in its early stages, we have committed ourselves more to its general presentation than to the precise parameter settings. In the near future, we therefore aim to investigate more fully the possible variation that the method allows. First, we will focus on the implementation of the word space model, by studying word-based models with other context sizes as well as syntax-based approaches. Second, we want to examine other ways in which the word-based model and the classic keyword approach can be combined, apart from the multiplication of ranks that we have proposed here. While this large freedom in parameter settings could be seen as a weakness of the proposed method, the fact that we obtained similar results for all settings we have tried out so far, adds to our confidence that word space models present a sensible complementation of the classic keyword approaches, irrespective of the precise parameter settings.

In addition to those modelling issues, there are a number of other extensions we would like to explore. First, the Gold Standard we have used so far is rather limited in scope. We therefore plan to incorporate more sources on language variation to test the robustness of our approach. Finally, as we have observed a number of times, the method in its present form is not sensitive to possibly infrequent meanings of a polysemous word. This may be solved by the application of a clustering approach that is able to cluster a word's contexts into several sense clusters (Schütze, 1998). Still, the promising results in this paper encourage us to believe that the current approach has a future as a new method in language-variational research and as a tool for lexicography.

References

John A. Bullinaria and Joseph P. Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behaviour Research Methods*, 39:510–526.

Ton Den Boon and Dirk Geeraerts. 2005. *Van Dale Groot Woordenboek van de Nederlandse taal (14e ed.)*. Van Dale Lexicografie, Utrecht/Antwerp.

Ted Dunning. 1993. Accurate methods for the statis-

tics of surprise and coincidence. *Computational Linguistics*, 19:61–74.

Dirk Geeraerts, Stefan Grondelaers, and Dirk Speelman. 1999. *Convergentie en Divergentie in de Nederlandse Woordenschat*. Meertens Instituut, Amsterdam.

Adam Kilgarriff. 2001. Comparing corpora. *International Journal of Corpus Linguistics*, 6(1):1–37.

Willy Martin. 2005. Het Belgisch-Nederlands anders bekeken: het Referentiebestand Belgisch-Nederlands (RBBN). Technical report, Vrije Universiteit Amsterdam, Amsterdam, Holland.

Sebastian Padó and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.

Yves Peirsman. 2008. Word space models of semantic similarity and relatedness. In *Proceedings of the 13th ESSLLI Student Session*, pages 143–152.

Reinhard Rapp. 1999. Automatic identification of word translations from unrelated English and German corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 519–526, College Park, Maryland.

Paul Rayson, Damon Berridge, and Brian Francis. 2004. Extending the cochran rule for the comparison of word frequencies between corpora. In *Proceedings of the 7ièmes Journées Internationales d'Analyse Statistique des Données Textuelles (JADT 2004)*, pages 926–936, Louvain-la-Neuve, Belgium.

Magnus Sahlgren. 2006. *The Word-Space Model. Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations Between Words in High-dimensional Vector Spaces*. Ph.D. thesis, Stockholm University, Stockholm, Sweden.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–124.

Mike Scott. 1997. PC analysis of key words – and key key words. *System*, 25(2):233–245.

Dirk Speelman, Stefan Grondelaers, and Dirk Geeraerts. 2006. A profile-based calculation of region and register variation: The synchronic and diachronic status of the two main national varieties of Dutch. In Andrew Wilson, Dawn Archer, and Paul Rayson, editors, *Corpus Linguistics around the World*, pages 195–202. Rodopi, Amsterdam.

Dirk Speelman, Stefan Grondelaers, and Dirk Geeraerts. 2008. Variation in the choice of adjectives in the two main national varieties of Dutch. In Gitte Kristiansen and René Dirven, editors, *Cognitive Sociolinguistics. Language Variation, Cultural Models, Social Systems*, pages 205–233. Mouton de Gruyter, Berlin.

Unsupervised Classification with Dependency Based Word Spaces

Klaus Rothenhäusler and Hinrich Schütze
Institute for Natural Language Processing
University of Stuttgart
Stuttgart, Germany
{Klaus.Rothenhaeusler, Hinrich.Schuetze}@ims.uni-stuttgart.de

Abstract

We present the results of clustering experiments with a number of different evaluation sets using dependency based word spaces. Contrary to previous results we found a clear advantage using a parsed corpus over word spaces constructed with the help of simple patterns. We achieve considerable gains in performance over these spaces ranging between 9 and 13% in absolute terms of cluster purity.

1 Introduction

Word space models have become a mainstay in the automatic acquisition of lexical semantic knowledge. The computation of semantic relatedness of two words in such models is based on their distributional similarity. The most crucial way in which such models differ is the definition of distributional similarity: In a regular word space model the observed distribution concerns the immediate neighbours of a word within a predefined window to the left and right (Schütze, 1992; Sahlgren, 2006). Early on in the development as an alternative models were proposed that relied on the similarity of the distribution of syntactic relations (Hindle, 1990; Padó and Lapata, 2007). More recently the distribution of the occurrence within simple patterns defined in the form of regular expressions that are supposed to capture explicit semantic relations was explored as the basis of distributional similarity (Almuhareb and Poesio, 2004).

Whereas dependency based semantic spaces have been shown to surpass other word space models for a number of problems (Padó and Lapata, 2007; Lin, 1998), for the task of categorisation simple pattern based spaces have been shown to perform equally good if not better (Poesio and Almuhareb, 2005b; Almuhareb and Poesio, 2005b). We want to show that dependency based spaces also fare better in these tasks if the dependency relations used are selected reasonably. At the same time we want to show that such a system can be built with freely available components and without the need to rely on the index of a proprietary search engine vendor.

We propose to use the web acquired data of the ukWaC (Ferraresi et al., 2008), which is huge but still manageable and comes in a pre-cleaned version with HTML markup removed. It can easily be fed into a parser like MiniPar which allows for the subsequent extraction of dependency relations of different types and complexity. In particular we work with dependency paths that can reach beyond direct dependencies as opposed to Lin (1998) but in the line of Pado and Lapata (2007). In contrast to the latter, however, different paths that end in the same word are not generally mapped to the same dimension in our model. A path in a dependency graph can pass through several nodes and encompass different relations.

We experimented with two sets of nouns previously used in the literature for word clustering. The nouns in both sets are taken from a number of different WordNet categories. Hence, the task consists in clustering together the words from the same category. By keeping the clustering algorithm constant, differences in performance can be attributed to the differences of the word representations.

The next section provides a formal description of our word space model. Section 3 reports on our clustering experiments with two sets of concepts used previously to evaluate the categorisation abilities of word spaces. Section 4 discusses these re-

Proceedings of the EACL 2009 Workshop on GEMS: GEometical Models of Natural Language Semantics, pages 17–24,
Athens, Greece, 31 March 2009.

sults and draws some conclusions.

2 Word Space Construction

We follow the formalisation and terminology developed in Pado and Lapata (2007) according to which a dependency based space is determined by the sets of its basis elements B and targets T that form a matrix $M = B \times T$, a similarity function S that assigns a real-valued similarity measure to pairs of elements from T, the association measure A that captures the strength of the relation between a target and a basis element, the context selection function *cont*, the basis mapping function μ and the path value function v. Our set of targets is always a subset of the lemmas output by MiniPar. The remaining elements are defined in this section. We use π to denote a path in a dependency graph which is conceived of as an undirected graph for this purpose. So, in general a dependency path has an upward and downward part where one can have length zero. All the paths used to define the contexts for target words are anchored there, i.e. they start from the target.

In choosing the context definitions that determine what dependency paths are used in the construction of the word vectors, we oriented ourselves at the sets proposed in Pado and Lapata (2007). As Pado and Lapata (2007) achieved their best results with it we started from their medium sized set of context definitions, from which we extracted the appropriate ones for our experiments and added some that seemed to make sense for our purposes: As our evaluation sets consist entirely of nouns, we used only context definitions that start at a noun. Thereby we can ensure that only nominal uses are recorded in a word vector if a target word can have different parts of speech. The complete set of dependency relations our context selection function *cont* comprises is given in Figure 1 along with an example for each.

We only chose paths that end in an open word class assuming that they are more informative about the meaning of a target word. Paths ending in a preposition for instance, as used by Pado and Lapata (2007), were not considered. For the same reason we implemented a simple stop word filter that discards paths ending in a pronoun, which are assigned the tag `N` by MiniPar just like any other noun.

On the other hand we added the relation between a prepositional complement and the noun it modifies (appearing as relation IX in Figure 1) as a close approximation of the pattern used by (Almuhareb and Poesio, 2004) to identify attributes of a concept as detailed in the next section. Path specifications X and XI are also additions we made that are thought to gather additional attribute values to the ones already covered by III.

As a basis mapping function μ we used a generalisation of the one used by Grefenstette (1994) and Lin (1998). They map a dependency between two words to a pair consisting of the relation label l and the end word of the dependency $end(\pi)$. As we use paths that span more than a single relation, this approach is not directly applicable to our setup. Instead we use a mapping function that maps a path to the sequence of edge labels through which it passes combined with the end word:

$$\mu(\pi) = (l(\pi), end(\pi))$$

where $l(\cdot)$ is a labelling function that returns the sequence of edge labels for a given path. With this basis mapping function the nodes or words respectively through which a path passes are all neglected except for the node where the path ends. So, for the noun *human* the sequence *human and mouse genome* as well as the sequence *human and chimpanzee genome* increase the count for the same basis element `:N:conj:N:*:N:nn:N:genome`. Here we use a path notation of the general form:

$$(:\texttt{POS}:\texttt{rel}:\texttt{POS}:\{\texttt{word},*\})^n$$

where `POS` is a part of speech, `rel` a relation and `word` a node label, i.e. a lemma, all as produced by MiniPar. The length of a path is determined by n and the asterisk (`*`) indicates that a node label is ignored by the basis mapping function.

As an alternative we experimented with a lexical basis mapping function that maps a path to its end word:

$$\mu(\pi) = end(\pi)$$

This reduces the number of dimensions considerably and yields semantic spaces that are similar to window based word spaces. As this mapping function consistently delivered worse results, we dropped it from our evaluation.

Considering that (Padó and Lapata, 2007) only reported very small differences for different path valuation functions, we only used a constant valuation of paths:

$$v_{\text{const}}(\pi) = 1$$

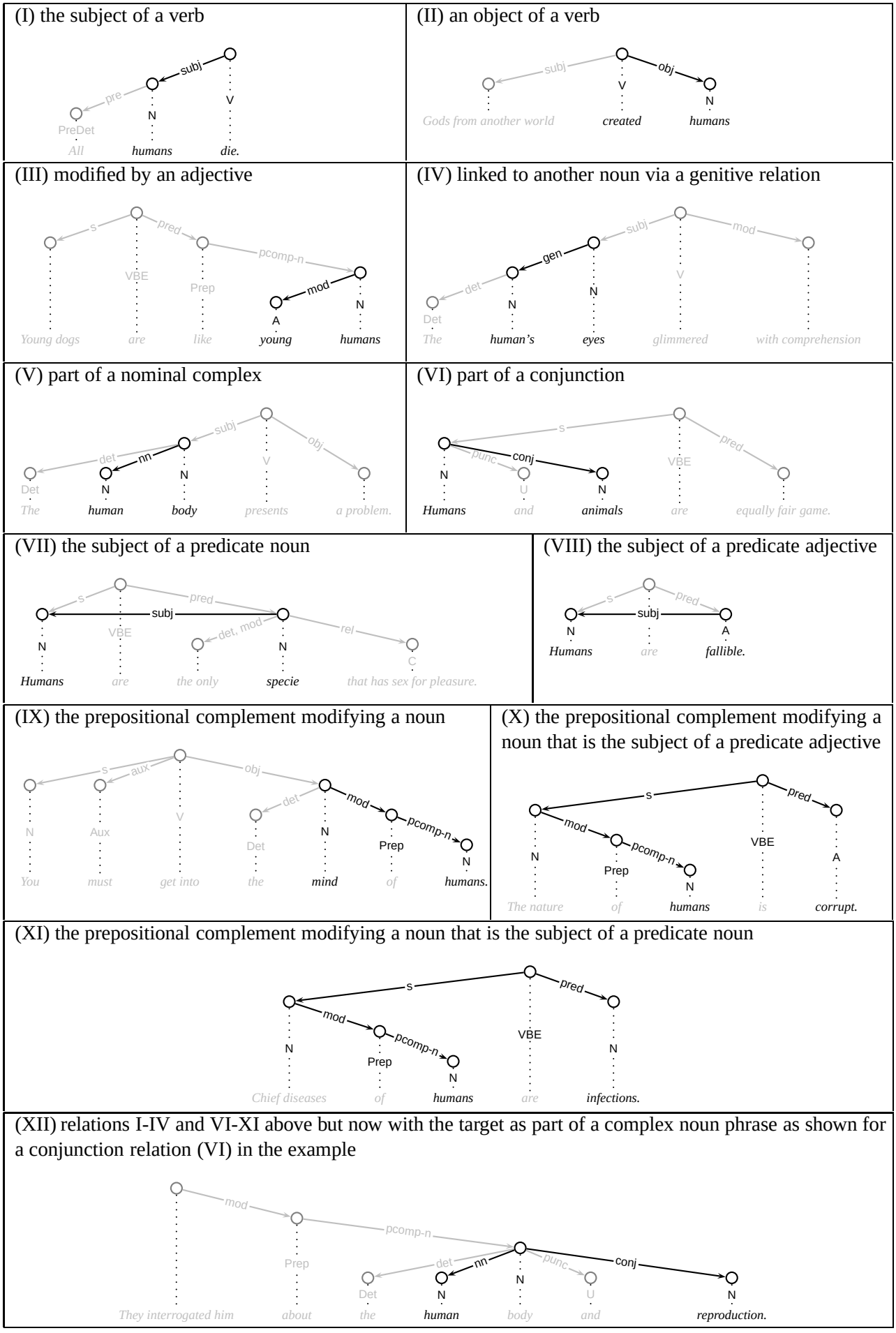


Figure 1: Context definitions used in the construction of our word spaces. All examples show contexts for the target *human*. Greyed out parts are just for illustrative purposes and have no impact on the word vectors. The examples are slightly simplified versions of sentences found in ukWaC.

Thus, an occurrence of any path, irrespective of length or grammatical relations that are involved, increases the count of the respective basis element by one.

We implemented three different association functions, A, to transform the raw frequency counts and weight the influence of the different co-occurrences. We worked with an implementation of the *log likelihood ratio* (g-Score) as proposed by Dunning (1993) and two variants of the *t-score*, one considering all values (t-score) and one where only positive values (t-score$^+$) are kept following the results of Curran and Moens (2002). We also experimented with different frequency cutoffs removing dimensions that occur very frequently or very rarely.

3 Evaluation

For all our experiments we used the ukWaC corpus[1] to construct the word spaces, which was parsed using MiniPar. The latter provides lemma information, which we used as possible target and context words. The word vectors we built from this data were represented as pseudo documents in an inverted index. To our knowledge the experiments described in this paper are the first to work with a completely parsed version of the ukWaC.

For the evaluation the word vectors for the test sets were clustered into a predefined number of clusters corresponding to the number of concept classes from which the words were drawn. All experiments were conducted with the CLUTO toolkit (Karypis, 2003) using the repeated bisections clustering algorithm with global optimisation and the cosine as a distance measure to maintain comparability with related work, e.g. Baroni et al. (2008).

As the main evaluation measure we used purity for the whole set as supplied by CLUTO. For a clustering solution Ω of n clusters and a set of classes C, purity can be defined as:

$$\text{purity}(\Omega, C) = \frac{1}{n}\sum_{k}\max_{j}|\omega_k \cap c_j|$$

where ω_k denotes the set of terms in a cluster and c_j the set of terms in a class. This aggregate measure of purity corresponds to the weighted sum of purities for the individual clusters, which is defined as the ratio of items in a cluster that belong to the majority class. The results for the two test sets we used are described in the following two subsections.

3.1 Results for 214 nouns from Almuhareb and Poesio (2004)

The first set we worked with was introduced by Almuhareb and Poesio (2004) and consists of 214 nouns from 13 different categories in WordNet. In the original paper the best results were achieved with vector representations built from concept attributes and their values as identified by simple patterns. For the identification of attribute values of a concept C the following pattern was used

"[a|an|the] * C [is|was]"

It will find instances such as *an adult human is* identifying *adult* as a value for an attribute (age) of [HUMAN] (we use small capitals enclosed in square brackets to denote a concept). Attributes themselves are searched with the pattern

"the * of the C [is|was]"

A match for the concept [HUMAN] would be *the dignity of the human is*, which yields *dignity* as an attribute. These patterns were translated into queries and submitted to the Google[2] search engine.

We compare our dependency based spaces with the results achieved with the pattern based approach in Table 1.

association measure	g-score	t-score	t-score$^+$
dependency based space	77.1%	85.5%	96.7%
window based space	84.1%	82.7%	89.3%
pattern based space	-	-	85.5%

Table 1: Categorisation results for the 214 concepts and 13 classes proposed in Almuhareb and Poesio (2004), which is also the source of the result for the pattern based space. They only used t-score$^+$. The numbers given are the best accuracies achieved under the different settings.

For the window based space we used the best performing in a free association task with a window size of six words to each side and all the

[1] http://wacky.sslmit.unibo.it

[2] http://www.google.com

context	accuracy	# dimensions
(I)	82.2%	7359
(II)	92.5%	6680
(III)	88.3%	45322
(IV)	–	37231
(V)	82.2%	240157
(VI)	95.3%	93917
(VII)	86.9%	45527
(VIII)	77.1%	5245
(IX)	91.6%	87765
(X)	–	2186
(XI)	–	6967
(XII)	93.0%	188763

Table 2: Clustering results using only one kind of path specification. For (IV), (X) and (XI) purity values are missing because vectors for some of the words could not be built.

removed context	accuracy
(I)	97.2%
(II)	97.7%
(III)	97.2%
(IV)	97.2%
(V)	98.1%
(VI)	96.3%
(VII)	97.2%
(VIII)	97.2%
(IX)	96.7%
(X)	97.2%
(XI)	97.2%
(XII)	96.7%

Table 3: Clustering results for spaces with one context specification removed.

words that appeared at least two times as dimensions ignoring stop words. The effective dimensionality of the so built word vectors is 417 837.

The results for the dependency based spaces were built by selecting all paths without any frequency thresholds which resulted in a set of 767 119 dimensions.

As can be seen, both window and dependency based spaces exceed the pattern based space for certain association measures. But the dependency space also has a clear advantage over the window based space. In particular the t-score$^+$ measure yields very good results. In contrast the g-score offers the worst results with the t-score retaining negative values somewhere in between. For our further experiments we hence used the t-score$^+$ association measure.

3.1.1 Further Analysis

We ran a number of experiments to quantify the impact the different kinds of paths have on the clustering result. We first built spaces using only a single kind of path to find out how good each performs on its own. The result can be found in Table 2. For some of the words in the evaluation set no contexts could be found when only one of the two most complex context specifications (X), (XI) was used or when the context was reduced to the genitive relation (IV). Apart from that the results suggest that even a single type of relation on its own can prove highly effective. Especially the conjunctive relation (VI) performs very well with a purity value of 95.3%.

To further clarify the role of the different kinds of contexts, we ran the experiment with word spaces where we removed each one of the twelve context specifications in turn. The results as given in Table 3 are a bit astonishing at first sight: Only the removal of the conjunctive relation actually leads to a decrease in performance. All the other contexts seem to be either redundant – with performance staying the same when they are removed – or even harmful – with performance increasing once they are removed. Having observed this, we tried to remove further context specifications and surprisingly found that the best performance of 98.1% can be reached by only including the conjunction (VI) and the object (II) relations. The dimensionality of these vectors is only a fraction of the original ones with 100 597.

The result for the best performing dependency based space listed in the table is almost perfect. Having a closer look at the results reveals that in fact only four words are put into a wrong cluster. These words are: *lounge, pain, mouse, oyster*.

The first is classified as [BUILDING] instead of [FURNITURE]. In the case of *lounge* the misclassification seems to be attributable to the ambiguity of the word which can either denote a piece of furniture or a waiting room. The latter is apparently the more prominent sense in the data. In this usage the word often appears in conjunctions with *room* or *hotel* just like *restaurant, inn* or *clubhouse*.

Pain is misclassified as an [ILLNESS] instead of a [FEELING] which is at least a close miss. The misclassification of *mouse* as a [BODY PART] seems rather odd on the other hand. The reason for

it becomes apparent when looking at the most descriptive and discriminating features of the [BODY PART] cluster: In both lists the highest in the ranking is the dimension `:N:mod:A:left`, i.e. *left* as an adjectival modifier of the word in question. The prominence of this particular modification is of course due to the fact that a lot of body parts come in pairs and that the members of these pairs are commonly identified by assigning them to the left or right half of the body. Certainly, the word *mouse* enters this cluster not through its sense of *mouse*[1] as an animal but rather through its sense of *mouse*[2] as a piece of computer equipment that has two buttons, which are also referred to as the left and right one. Unfortunately, MiniPar frequently resolves *left* in a wrong way as a modifier of *mouse* instead of *button*.

Finally for *oyster* which is put into the [EDIBLE FRUIT] instead of the [ANIMAL] cluster it is conspicuous that *oyster* is the only sea animal in the evaluation set and consequently it rarely occurs in conjunctions with the other animals. Conjunctions, however, seem to be the most important features for defining all the clusters. Additionally *oyster* scores low on a lot of dimensions that are typical for a big number of the members of the animal cluster, e.g. `:N:obj:V:kill`.

3.2 Results for 402 words from Almuhareb and Poesio (2005a)

In Poesio and Almuhareb (2005a) a larger evaluation set is introduced that comprises 402 nouns sampled from the hierarchies under the 21 unique beginners in WordNet. The words were also chosen so that candidates from different frequency bands and different levels of ambiguity were represented. Further results using this set are reported in Almuhareb and Poesio (2005b). The best result was obtained with the attribute pattern alone and filtering to include only nouns. We tried to assemble word vectors with the same patterns based on the ukWaC corpus. But even if we included both patterns, we were only able to construct vectors for 363 of the 402 words. For 118 of them the number of occurrences, on which they were based, was less than ten. This gives an impression of the size of the index that is necessary for such an approach. To date such an immense amount of data is only available through proprietary search engine providers. This makes a system dependant upon the availability of an API of such a vendor. In fact the version of the Google API on which the original experiments relied has since been axed. Our approach circumvents such problems.

We ran analogous experiments to the ones described in the previous section on this evaluation set, now producing 21 clusters. The results given in Table 4 are for a dependency space without any frequency thresholds and the complete set of context specifications as defined above. The settings for the window based space were also the same (6 words to each side). Again the results achieved with the t-score$^+$ association were clearly superior to the others and were used in all the following experiments. Unsurprisingly, for this more difficult task the performance is not as good as for the smaller set but nevertheless the superiority of the dependency based space is clearly visible with an absolute increase in cluster purity of 8.2% compared with the pattern based space.

association measure	g-score	t-score	t-score$^+$
dependency based space	67.9%	67.2%	79.1%
window based space	65.7%	60.7%	67.9%
pattern based space	-	-	70.9%

Table 4: Categorisation results for the 402 concepts and 21 classes proposed in Almuhareb and Poesio (2005a) which is also the source of the result for the pattern based space. The numbers given are the best accuracies achieved under the different settings.

3.2.1 Further Analysis

Again we ran further experiments to determine the impact of the different kinds of relations. The removal of any single context specification leads to a performance drop with this evaluation set. The smallest decrease is observed when removing context specification XII. However, as we had seen in the previous experiment with the smaller set that only two context specifications suffice to reach peak performance, we conducted another experiment where we started from the best performing space constructed from a single context specification (the conjunction relation, VI) and successively added the specification that led to the biggest performance gain. The crucial results are

majority class	concepts
solid	tetrahedron, salient, ring, ovoid, octahedron, knob, icosahedron, fluting, dome, dodecahedron, cylinder, cuboid, cube, crinkle, concavity, *samba, coco, nonce, divan, ball, stitch, floater, trove, hoard, mouse*
time	yesteryear, yesterday, tonight, tomorrow, today, quaternary, period, moment, hereafter, gestation, future, epoch, day, date, aeon, *stretch, snap, throb, straddle, nap*
motivation	wanderlust, urge, superego, obsession, morality, mania, life, impulse, ethics, dynamic, conscience, compulsion, *plasticity, opinion, acceptance, sensitivity, desire, interest*
assets	wager, taxation, quota, profit, payoff, mortgage, investment, income, gain, fund, credit, capital, allotment, allocation, *possession, inducement, incentive, disincentive, deterrence, share, sequestrian, cheque, check, bond, tailor*
district	village, town, sultanate, suburb, state, shire, seafront, riverside, prefecture, parish, metropolis, land, kingdom, county, country, city, canton, borough, borderland, anchorage, *tribe, nation, house, fen, cordoba, faro*
legal document	treaty, statute, rescript, obligation, licence, law, draft, decree, convention, constitution, bill, assignment, *commencement, extension, incitement, caliphate, clemency, venture, dispensation*
physical property	weight, visibility, temperature, radius, poundage, momentum, mass, length, diameter, deflection, *taper, indentation, droop, corner, concavity*
social unit	troop , team, platoon, office, legion, league, household, family, department, confederacy, company, committee, club, bureau, brigade, branch, agency
atmospheric phenomenon	wind, typhoon, tornado, thunderstorm, snowfall, shower, sandstorm, rainstorm, lightning, hurricane, fog, drizzle, cyclone, crosswind, cloudburst, cloud, blast, aurora, airstream, *glow*
social occasion	wedding, rededication, prom, pageantry, inaugural, graduation, funeral, fundraiser, fiesta, fete, feast, enthronement, dance, coronation, commemoration, ceremony, celebration, *occasion, raffle, beano*
monetary unit	zloty, yuan, shilling, rupee, rouble, pound, peso, penny, lira, guilder, franc, escudo, drachma, dollar, dirham, dinar, cent
tree	sycamore, sapling, rowan, pine, palm, oak, mangrove, jacaranda, hornbeam, conifer, cinchona, casuarina, acacia, *riel*
chemical element	zinc, titanium, silver, potassium, platinum, oxygen, nitrogen, neon, magnesium, lithium, iron, hydrogen, helium, germanium, copper, charcoal, carbon, calcium, cadmium, bismuth, aluminium, *gold*
illness	smallpox, plague, meningitis, malnutrition, leukemia, hepatitis, glaucoma, flu, eczema, diabetes, cirrhosis, cholera, cancer, asthma, arthritis, anthrax, acne, *menopause*
feeling	wonder, shame, sadness, pleasure, passion, love, joy, happiness, fear, anger, *heaviness, coolness, torment, tenderness, suffering, stinging*
vehicle	van, truck, ship, rocket, pickup, motorcycle, helicopter, cruiser, car, boat, bicycle, automobile, airplane, aircraft, *jag*
creator	producer, photographer, painter, originator, musician, manufacturer, maker, inventor, farmer, developer, designer, craftsman, constructor, builder, artist, architect, *motivator*
pain	toothache, soreness, sting, soreness, sciatica, neuralgia, migraine, lumbago, headache, earache, burn, bellyache, backache, ache, *rheumatism, pain*
animal	zebra, turtle, tiger, sheep, rat, puppy, monkey, lion, kitten, horse, elephant, dog, deer, cow, cat, camel, bull, bear
game	whist, volleyball, tennis, softball, soccer, rugby, lotto, keno, handball, golf, football, curling, chess, bowling, basketball, baccarat, *twister*
edible fruit	watermelon, strawberry, pineapple, pear, peach, orange, olive, melon, mango, lemon, kiwi, grape, cherry, berry, banana, apple, *oyster, walnut, pistachio, mandarin, lime, fig, chestnut*

Figure 2: Optimal clustering for large evaluation set.

contexts used	purity
(VI)	73.4%
(VI), (II)	76.6%
(VI), (II), (III)	80.1%

Table 5: Clustering the larger evaluation set with an increasing number of context specifications.

given in Table 5. As can be seen the object relation is added first again. This time though the inclusion of adjectival modification brings another performance increase which is even one per cent above the result for the space built from all possible relations. The addition of any further contexts consistently degrades performance. The clustering solution thus produced is given in Figure 2. From the 1 872 698 dimension used in the original space only 341 214 are retained.

4 Discussion and Conclusion

Our results are counterintuitive at first sight as it could be expected that a larger number of different contexts would increase performance. Instead we see the best performance with only a very lim-

ited set of possible contexts. We suspect that this behaviour is due to a large amount of correlation between the different kinds of contexts. The addition of further contexts beyond a certain point therefore has no positive effect. As an indication for this it might be noticed that the three context specifications that yield the best result for the 402 word set comprise relations with the three main open word classes. It is to be expected that they contribute orthogonal information that covers central dimensions of meaning. The slight decrease in performance that can be observed when further contexts are added is probably due to chance fluctuations and almost certainly not significant; with significance being hard to determine for any of the results.

However, it is obviously necessary to cover a basic variety of features. Patterns which are used to explicitly track semantic relations on the textual surface seem to be too restrictive. Information accessible from co-occurring verbs for example is completely lost. In a regular window based word space such information is retained and its performance is competitive with a pattern based approach. This method is obviously too liberal, though, if compared to the dependency spaces.

In general we were able to show that semantic spaces are obviously able to capture categorical knowledge about concepts best when they are built from a syntactically annotated source. This is true even if the context specification used is not the most parsimonious. The problem of determining the right set of contexts is therefore rather an optimisation issue than a question of using dependency based spaces or not. It is a considerable one, though, as computations are much cheaper with vectors of reduced dimensionality, of course.

For the categorisation task the inclusion of more complex relations reaching over several dependencies does not seem to be helpful considering they can all be dropped without a decrease in performance. As Pado and Lapata (2007) reached better results in their experiments with a broader set of context specifications we conclude that the selection of the kinds of context to include when constructing a word space depends largely on the task at hand.

References

A. Almuhareb and M. Poesio. 2004. Attribute-based and value-based clustering: An evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 158–165, Barcelona, Spain, July. Association for Computational Linguistics.

M. Poesio and A. Almuhareb. 2005a. Concept learning and categorization from the web. In *Proceedings of CogSci2005 - XXVII Annual Conference of the Cognitive Science Society*, pages 103–108, Stresa, Italy.

A. Almuhareb and M. Poesio. 2005b. Finding attributes in the web using a parser. In *Proceedings of Corpus Linguistics*, Birmingham.

Marco Baroni, Stefan Evert, and Alessandro Lenci, editors. 2008. *ESSLLI Workshop on Distributional Lexical Semantics*, Hamburg, August.

J. R. Curran and M. Moens. 2002. Improvements in automatic thesaurus extraction. In *Proceedings of the ACL-02 workshop on Unsupervised lexical acquisition*, pages 59–66, Morristown, NJ, USA. Association for Computational Linguistics.

T. Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

A. Ferraresi, E. Zanchetta, M. Baroni, and S. Bernardini. 2008. Introducing and evaluating ukwac, a very large web-derived corpus of english. In *Proceedings of the WAC4 Workshop at LREC 2008*.

G. Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, Dordrecht.

D. Hindle. 1990. Noun classification from predicate-argument structures. In *Meeting of the Association for Computational Linguistics*, pages 268–275.

G. Karypis. 2003. Cluto: A clustering toolkit. technical report 02-017. Technical report, University of Minnesota, November.

D. Lin. 1998. Automatic retrieval and clustering of similar words. In *COLING-ACL*, pages 768–774.

S. Padó and M. Lapata. 2007. Dependency-based construction of semantic space models. *Comput. Linguist.*, 33(2):161–199.

M. Poesio and A. Almuhareb. 2005b. Identifying concept attributes using a classifier. In *Proceedings of the ACL-SIGLEX Workshop on Deep Lexical Acquisition*, pages 18–27, Ann Arbor, Michigan, June. Association for Computational Linguistics.

M. Sahlgren. 2006. *The Word Space Model*. Ph.D. thesis, Department of Linguistics, Stockholm University.

H. Schütze. 1992. Dimensions of meaning. In *Supercomputing '92: Proceedings of the 1992 ACM/IEEE conference on Supercomputing*, pages 787–796, Los Alamitos, CA, USA. IEEE Computer Society Press.

A Study of Convolution Tree Kernel with Local Alignment

Lidan Zhang
Department of Computer Science
HKU, Hong Kong
lzhang@cs.hku.hk

Kwok-Ping Chan
Department of Computer Science
HKU, Hong Kong
kpchan@cs.hku.hk

Abstract

This paper discusses a new convolution tree kernel by introducing local alignments. The main idea of the new kernel is to allow some syntactic alternations during each match between subtrees. In this paper, we give an algorithm to calculate the composite kernel. The experiment results show promising improvements on two tasks: semantic role labeling and question classification.

1 Introduction

Recently kernel-based methods have become a state-of-art technique and been widely used in natural language processing applications. In this method, a key problem is how to design a proper kernel function in terms of different data representations. So far, there are two kinds of data representations. One is to encode an object with a flat vector whose element correspond to an extracted feature from the object. However the feature vector is sensitive to the structural variations. The extraction schema is heavily dependent on different problems. On the other hand, kernel function can be directly calculated on the object. The advantages are that the original topological information is to a large extent preserved and the introduction of additional noise may be avoided. Thus structure-based kernels can well model syntactic parse tree in a variety of applications, such as relation extraction(Zelenko et al., 2003), named entity recognition(Culotta and Sorensen, 2004), semantic role labeling(Moschitti et al., 2008) and so on.

To compute the structural kernel function, Haussler (1999) introduced a general type of kernel function, called“ Convolution kernel”. Based on this work, Collins and Duffy (2002) proposed a tree kernel calculation by counting the common subtrees. In other words, two trees are considered if and only if these two trees are exactly same. In real sentences, some structural alternations within a given phrase are permitted without changing its usage. Therefore, Moschitti (2004) proposed partial trees to partially match between subtrees. Kashima and Koyanagi (2002) generalize the tree kernel to labeled order tree kernel with more flexible match. And from the idea of introducing linguistical knowledge, Zhang et al. (2007) proposed a grammar-driven tree kernel, in which two subtrees are same if and only if the corresponding two productions are in the same manually defined set. In addition, the problem of hard matching can be alleviated by processing or mapping the trees. For example, Tai mapping (Kuboyama et al., 2006) generalized the kernel from counting subtrees to counting the function of mapping. Moreover multi-source knowledge can benefit kernel calculation, such as using dependency information to dynamically determine the tree span (Qian et al., 2008).

In this paper, we propose a tree kernel calculation algorithm by allowing variations in productions. The variation is measured with local alignment score between two derivative POS sequences. To reduce the computation complexity, we use the dynamic programming algorithm to compute the score of any alignment. And the top n alignments are considered in the kernel.

Proceedings of the EACL 2009 Workshop on GEMS: GEometical Models of Natural Language Semantics, pages 25–32,
Athens, Greece, 31 March 2009.

Another problem in Collins and Duffy's tree kernel is context-free. It does not consider any semantic information located at the leaf nodes of the parsing trees. To lexicalized tree kernel, Bloehdorn et al. (2007) considered the associated term similarity by virtue of WordNet. Shen et al. (2003) constructed a separate lexical feature containing words on a given path and merged into the kernel in linear combination.

The paper is organized as follows. In section 2, we describe the commonly used tree kernel. In section 3, we propose our method to make use of the local alignment information in kernel calculation. Section 4 presents the results of our experiments for two different applications (Semantic Role Labeling and Question Classification). Finally section 5 provides our conclusions.

2 Convolution Tree Kernel

The main idea of tree kernel is to count the number of common subtrees between two trees T_1 and T_2. In convolutional tree kernel (Collins and Duffy, 2002), a tree(T) is represented as a vector $h(T) = (h_1(T), ..., h_i(T), ..., h_n(T))$, where $h_i(T)$ is the number of occurrences of the i^{th} tree fragment in the tree T. Since the number of subtrees is exponential with the parse tree size, it is infeasible to directly count the common subtrees. To reduce the computation complexity, a recursive kernel calculation algorithm was presented. Given two trees T_1 and T_2,

$$\begin{aligned} K(T_1, T_2) &= < h(T_1), h(T_2) > \\ &= \sum_i h_i(T_1) h_i(T_2) \\ &= \sum_i (\sum_{n_1 \in N_{T_1}} I_i(n_1) \sum_{n_2 \in N_{T_2}} I_i(n_2)) \\ &= \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \triangle(n_1, n_2) \end{aligned} \quad (1)$$

where, N_{T_1} and N_{T_2} are the sets of all nodes in trees T_1 and T_2, respectively. $I_i(n)$ is the indicator function to be 1 if i-th subtree is rooted at node n and 0 otherwise. And $\triangle(n_1, n_2)$ is the number of common subtrees rooted at n_1 and n_2. It can be computed efficiently according to the following rules:

(1) If the productions at n_1 and n_2 are different, $\triangle(n_1, n_2) = 0$

(2) If the productions at n_1 and n_2 are same, and n_1 and n_2 are pre-terminals, then $\triangle(n_1, n_2) = \lambda$

(3) Else, $\triangle(n_1, n_2) = \lambda \prod_j^{nc(n_1)} (1 + \triangle(ch(n_1, j), ch(n_2, j)))$

where $nc(n_1)$ is the number of children of n_1 in the tree. Note that $n_1 = n_2$ because the productions at n_1 and n_2 are same. $ch(n_1, j)$ represents the j^{th} child of node n_1. And $0 < \lambda \leq 1$ is the parameter to downweight the contribution of larger tree fragments to the kernel. It corresponds to $K(T_1, T_2) = \sum_i \lambda^{size_i} h_i(T_1) h_i(T_2)$, where $size_i$ is the number of rules in the i'th fragment. The time complexity of computing this kernel is $O(|N_{T_1}| \cdot |N_{T_2}|)$.

3 Tree Kernel with Local Alignment

3.1 General Framework

As we referred, one of problems in the basic tree kernel is its hard match between two rules. In other words, at each tree level, the two subtrees are required to be perfectly equal. However, in real sentences, some modifiers can be added into a phrase without changing the phrase's function. For example, two sentences are given in Figure 1. Considering "A1" role, the similarities between two subtrees(in circle) are 0 in (Collins and Duffy, 2002), because the productions "NP→DT ADJP NN" and "NP→DT NN" are not identical. From linguistical point of view, the adjective phrase is optional in real sentences, which does not change the corresponding semantic role. Thus the modifier components(like "ADJP" in the above example) should be neglected in similarity comparisons.

To make the hard match flexible, we can align two string sequences derived from the same node. Considering the above example,

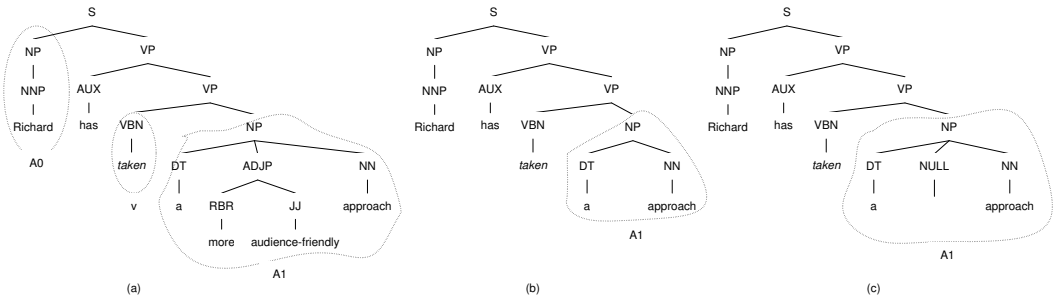


Figure 1: Syntactic parse tree with "A1" semantic role

an alignment might be "DT ADJP NN" vs "DT - NN", by inserting a symbol(-). The symbol(-) corresponds to a "NULL" subtree in the parser tree. And the "NULL" subtree can be regarded as a null character in the sentence, see Figure 1(c).

Convolution kernels, studied in (Haussler, 1999) gave the framework to construct a complex kernel from its simple elements. Suppose $x \in X$ can be decomposed into $x_1, ..., x_m \equiv \vec{x}$. Let R be a relation over $X_1 \times ... \times X_m \times X$ such that $R(\vec{x})$ is true iff $x_1, ..., x_m$ are parts of x. $R^{-1}(x) = \{\vec{x}|R(\vec{x}, x)\}$, which returns all components. For example, x is any string, then $\vec{x}$ can be its characters. The convolution kernel K is defined as:

$$K(x, y) = \sum_{\vec{x} \in R^{-1}(x), \vec{y} \in R^{-1}(y)} \prod_{d=1}^{m} K_d(x_d, y_d) \quad (2)$$

Considering our problem, for example, a derived string sequence x by the rule "$n_1 \rightarrow x$". $R(x_i, x)$ is true iff x_i appears in the right hand of x. Given two POS sequences x and y derived from two nodes n_1 and n_2, respectively, $A(x, y)$ denotes all the possible alignments of the sequence. The general form of the kernel with local alignment is defined as:

$$K'(n_1, n_2) = \sum_{(i,j) \in A(x,y)} K(n_1^i, n_2^j) \quad (3)$$

$$\triangle'(n_1, n_2) = \lambda \sum_{(i,j) \in A(x,y)} AS^{(i,j)} \prod_{d=1}^{nc(n_1,i)} (1 + \triangle'(ch(n_1, i, d), ch(n_2, j, d)))$$

where, (i, j) denotes the i^{th} and j^{th} variation for x and y, $AS^{(i,j)}$ is the score for alignment i and j. And $ch(n_1, i, d)$ selects the d^{th} subtree for the i^{th} aligned schema of node n_1.

It is easily to prove the above kernel is positive semi-definite, since the kernel $K(n_1^i, n_2^j)$ is positive semi-definite. The native computation is impractical because the number of all possible alignments($|A(x, y)|$) is exponential with respect to $|x|$ and $|y|$. In the next section, we will discuss how to calculate $AS^{(i,j)}$ for each alignment.

3.2 Local Alignment Kernel

The local alignment(LA) kernel was usually used in bioinformatics, to compare the similarity between two protein sequences(x and y) by exploring their alignments(Saigo et al., 2004).

$$K_{LA}(x, y) = \sum_{\pi \in A(x,y)} \exp^{\beta s(x,y,\pi)} \quad (4)$$

where $\beta \geq 0$ is a parameter, $A(x, y)$ denotes all possible local alignments between x and y, and $s(x, y, \pi)$ is the local alignment score for a given alignment schema π, which is equal to:

$$s(x, y, \pi) = \sum_{i=1}^{|\pi|} S(x_{\pi_1^i}, y_{\pi_2^i}) - \sum_{j=1}^{|\pi|-1} [g(\pi_1^{i+1} - \pi_1^i) + g(\pi_2^{i+1} - \pi_2^i)] \quad (5)$$

In equation(5), S is a substitution matrix, and g is a gap penalty function. The alignment score is the sum of the substitution score between the correspondence at the aligned position, minus the sum of the gap penalty for the

case that '-' symbol is inserted. In natural language processing, the substitution matrix can be selected as identity matrix and no penalty is accounted.

Obviously, the direct computation of the original K_{LA} is not practical. Saigo (2004) presented a dynamic programming algorithm with time complexity $O(|x|\cdot|y|)$. In this paper, this dynamic algorithm is used to compute the kernel matrix, whose element(i,j) is used as $AS^{(i,j)}$ measurement in equation(3).

3.3 Local Alignment Tree Kernel

Now we embed the above local alignment score into the general tree kernel computation. Equation(3) can be re-written into following:

$$\triangle'(n_1,n_2)=\lambda\sum_{\pi\in A(x,y)}(\exp^{\beta s(x,y,\pi)}\times \prod_{k=1}^{nc(n_1,i)}(1+\triangle'(ch(n_1,i,k),ch(n_2,j,k)))) \quad (6)$$

To further reduce the computation complexity, a threshold (ξ) is used to filter out alignments with low scores. This can help to avoid over-generated subtrees and only select the significant alignments. In other words, by using the threshold (ξ), we can select the salient subtree variations for kernels. The final kernel calculation is shown below:

$$\triangle'(n_1,n_2)=\lambda\sum_{\substack{\pi\in A(x,y)\\ s(x,y,\pi)>\xi}}(\varepsilon^{\beta s(x,y,\pi)}\times \prod_{k=1}^{nc(n_1,i)}(1+\triangle'(ch(n_1,i,k),ch(n_2,j,k)))) \quad (7)$$

After filtering, the kernel is still positive semi-definite. This can be easily proved using the theorem in (Shin and Kuboyama, 2008), since this subset selection is transitive. More specifically, if $s(x,y,\pi)>\xi \bigwedge s(y,z,\pi')>\xi$, then $s(x,z,\pi+\pi')>\xi$.

The algorithm to compute the local alignment tree kernel is given in algorithm 1. For any two nodes pair(x_i and y_j), the local alignment score $M(x_i,y_j)$ is assigned. In the kernel matrix calculation, the worst case occurs when the tree is balanced and most of the alignments are selected.

Algorithm 1 algorithm for local alignment tree kernel

Require: 2 nodes n_1,n_2 in parse trees;The productions are $n_1\rightarrow x_1,...,x_m$ and $n_2\rightarrow y_1,...,y_n$

```
return △'(n1, n2)
if n1 and n2 are not same then
    △'(n1, n2) = 0
else
    if both n1 and n2 are pre-terminals then
        △'(n1, n2) = 1
    else
        calculate kernel matrix by equation( 4)
        for each possible alignment do
            calculate △'(n1, n2) by equation(7)
        end for
    end if
end if
```

4 Experiments

4.1 Semantic Role Labeling

4.1.1 Experiment Setup

We use the CoNLL-2005 SRL shared task data(Carreras and Marquez, 2005) as our experimental data. It is from the Wall Street Journal part of the Penn Treebank, together with predicate-arguments information from the PropBank. According to the shared task, sections 02-21 are used for training, section 24 for development and section 23 as well as some data from Brown corpus are left for test. The data sets are described in Table 1.

		Sentences	Arguments
Training		39,832	239,858
Dev		1,346	8,346
Test	WSJ	1,346	8,346
	Brown	450	2,350

Table 1: Data sets statistics

Considering the two steps in semantic role labeling, i.e. semantic role identification and recognition. We assume identification has been done correctly, and only consider the semantic role classification. In our experiment, we focus on the semantic classes include 6 core (A0-A5), 12 adjunct(AM-) and 8 reference(R-) arguments.

In our implementation, SVM-Light-TK[1] (Moschitti, 2004) is modified. For SVM multi-classifier, the ONE-vs-ALL (OVA) strategy is selected. In all, we prepare the data for each semantic role (r) as following:

(1) Given a sentence and its correct full syntactic parse tree;

(2) Let P be the predicate. Its potential arguments A are extracted according to (Xue and Palmer, 2004)

(3) For each pair $< p, a > \in P \times A$: if a covers exactly the words of semantic role of p, put minimal subtree $< p, a >$ into positive example set (T_r^+); else put it in the negative examples (T_r^-)

In our experiments, we set $\beta = 0.5$.

4.1.2 Experimental Results

The classification performance is evaluated with respect to accuracy, precision(p), recall(r) and $F_1 = 2pr/(p + r)$.

	Accuracy(%)
(Collins and Duffy, 2002)	84.35
(Moschitti, 2004)	86.72
(Zhang et al., 2007)	87.96
Our Kernel	88.48

Table 2: Performance comparison between different kernel performance on WSJ data

[1]http://dit.unitn.it/ moschitt/Tree-Kernel.htm

	P(%)	R(%)	$F_{\beta=1}$
Development	81.03	68.91	74.48
WSJ Test	84.97	79.45	82.11
Brown Test	76.95	70.94	73.51
WSJ+Brown	82.98	75.40	79.01

WSJ	P(%)	R(%)	F
A0	81.28	83.90	82.56
A1	84.22	66.39	74.25
A2	77.27	62.36	69.02
A3	93.33	21.21	34.57
A4	82.61	51.35	63.33
A5	100.00	40.00	57.41
AM-ADV	74.21	56.21	63.92
AM-CAU	75.00	46.09	57.09
AM-DIR	57.14	16.00	25.00
AM-DIS	77.78	70.00	73.68
AM-EXT	75.00	53.10	62.18
AM-LOC	89.66	74.83	81.57
AM-MNR	84.62	48.20	61.41
AM-MOD	96.64	92.00	94.26
AM-NEG	99.30	95.30	97.26
AM-PNC	48.20	28.31	35.67
AM-PRD	50.00	30.00	37.50
AM-TMP	87.87	73.43	80.00
R-A0	81.08	67.80	73.85
R-A1	77.50	49.60	60.49
R-A2	58.00	42.67	49.17
R-AM-CAU	100.00	25.00	40.00
R-AM-EXT	100.00	100.00	100.00
R-AM-LOC	100.00	55.00	70.97
R-AM-MNR	50.00	25.00	33.33
R-AM-TMP	85.71	52.94	65.46

Table 3: top: overall performance result on data sets ; bottom: detail result on WSJ data

Table 2 compares the performance of our method and other three famous kernels on WSJ test data. We implemented these three methods with the same settings described in the papers. It shows that our kernel achieves the best performance with 88.48% accuracy. The advantages of our approach are: 1). the alignments allow soft syntactic structure match; 2). threshold can avoid over-generation and selected salient alignments.

Table 3 gives our performance on data sets and the detail result on WSJ test data.

Similarity	Definition
Wu and Palmer	$sim_{WUP}(c_1, c_2) = \frac{2dep(lso(c_1,c_2))}{d(c_1,lso(c_1,c_2))+d(c_2,lso(c_1,c_2))+2dep(lso(c_1,c_2))}$
Resnik	$sim_{RES}(c_1, c_2) = -\log P(lso(c_1, c_2))$
Lin	$sim_{LIN}(c_1, c_2) = \frac{2\log P(lso(c_1,c_2))}{\log P(c_1)+\log P(c_2)}$

Table 4: popular semantic similarity measurements

4.2 Question Classification

4.2.1 Semantic-enriched Tree Kernel

Another problem in the tree kernel (Collins and Duffy, 2002) is the lack of semantic information, since the match stops at the pre-terminals. All the lexical information is encoded at the leaf nodes of parsing trees. However, the semantic knowledge is important in some text applications, like Question Classification. To introduce semantic similarities between words into our kernel, we use the framework in Bloehdorn et al. (2007) and rewrite the rule (2) in the iterative tree kernel calculation(in section 2).

> (2) If the productions at n_1 and n_2 are same, and n_1 and n_2 are pre-terminals, then $\triangle(n_1, n_2) = \lambda\alpha k_w(w_1, w_2)$

where w_1 and w_2 are two words derived from pre-terminals n_1 and n_2, respectively, and the parameter α is to control the contribution of the leaves. Note that each preterminal has one child or equally covers one word. So $k_w(w_1, w_2)$ actually calculate the similarity between two words w_1 and w_2.

In general, there are two ways to measure the semantic similarities. One is to derive from semantic networks such as WordNet (Mavroeidis et al., 2005; Bloehdorn et al., 2006). The other way is to use statistical methods of distributional or co-occurrence (Ó Séaghdha and Copestake, 2008) behavior of the words.

WordNet[2] can be regarded as direct graphs semantically linking concepts by means of relations. Table 4 gives some similarity measures between two arbitrary concepts c_1 and c_2. For our application, the word-to-word similarity can be obtained by maximizing the corresponding concept-based similarity scores. In our implementation, we use WordNet::Similarity package[3](Patwardhan et al., 2003) and the noun hierarchy of WordNet.

In Table 4, dep is the length of path from a node to its global root, $lso(c_1, c_2)$ represents the lowest super-ordinate of c_1 and c_2. The detail definitions can be found in (Budanitsky and Hirst, 2006) .

As an alternative, Latent Semantic Analysis(LSA) is a technique. It calculates the words similarities by means of occurrence of terms in documents. Given a term-by-document matrix X, its singular value decomposition is: $X = U\Sigma V^T$, where Σ is a diagonal matrix with singular values in decreasing arrangement. The column of U are singular vectors corresponding to the individual singular value. Then the latent semantic similarity kernel of terms t_i and t_j is:

$$sim_{LSA} =< U_k^i (U_k^j)^T > \qquad (8)$$

where $U_k = I_k U$ is to project U onto its first k dimensions. I_k is the identity matrix whose first k diagonal elements are 1 and all the other elements are 0. And U_k^i is the i-th row of the matrix U_k. From equation (8), the LSA-based similarity between two terms is the inner product of the two projected vectors. The details of LSA can be found in (Cristianini et al., 2002; Choi et al., 2001).

4.2.2 Experiment Results

In this set of experiment, we evaluate different types of kernels for Question Classification(QC) task. The duty of QC is to categorize questions into different classes. In

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

[3]http://search.cpan.org/dist/WordNet-Similarity

Accuracy(%)		1000	2000	3000	4000	5500
BOW		77.1	83.3	87.2	87.3	89.2
TK		80.2	86.2	87.4	88.6	91.2
LATK		80.4	86.5	87.5	88.8	91.6
$\alpha = 1$	WUP	81.3	87.3	88.0	89.8	92.5
	RES	81.0	87.1	87.9	89.5	92.2
	LIN	81.1	87.0	88.0	89.3	92.4
	LSA($k = 50$)	80.8	86.9	87.8	89.3	91.7

Table 5: Classification accuracy of different kernels on different data sets

this paper we use the same dataset as introduced in(Li and Roth, 2002). The dataset is divided[4] into 5500 questions for training and 500 questions from TREC 20 for testing. The total training samples are randomly divided into 5 subsets with sizes 1,000, 2,000, 3,000, 4,000 and 5,500 respectively. All the questions are labeled into 6 coarse grained categories and 50 fine grained categories: Abbreviations (abbreviation and expansion), Entity (animal, body, color, creation, currency, medical, event, food, instrument, language, letter, plant, product, religion, sport, substance, symbol, technique, term, vehicle, word), Description (definition, description, manner, reason), Human (description, group, individual, title), Location (city, country, mountain, state) and Numeric (code, count, date, distance, money, order, percent, period, speed, temperature, size, weight).

In this paper, we compare the linear kernel based on bag-of-word (BOW), the original tree kernel (TK), the local alignment tree kernel (section 3, LATK) and its correspondences with LSA similarity and a set of semantic-enriched LATK with different similarity metrics.

To obtain the parse tree, we use Charniak parser[5] for every question. Like the previous experiment, SVM-Light-TK software and the OVA strategy are implemented. In all experiments, we use the default parameter in SVM(e.g. margin parameter) and set $\alpha = 1$. In LSA model, we set $k = 50$. Finally, we use multi-classification accuracy to evaluate the performance.

Table 5 gives the results of the experiments. We can see that the local alignment tree kernel increase the multi-classification accuracy of the basic tree kernel by about 0.4%. The introduction of semantic information further improves accuracy. Among WordNet-based metrics, "Wu and Palmer" metric achieves the best result, i.e. 92.5%. As a whole, the WordNet-based similarities perform better than LSA-based measurement.

5 Conclusion

In this paper, we propose a tree kernel calculation by allowing local alignments. More flexible productions are considered in line with modifiers in real sentences. Considering text related applications, words similarities have been merged into the presented tree kernel. These similarities can be derived from different WordNet-based metrics or document statistics. Finally experiments are carried on two different applications (Semantic Role Labeling and Question Classification).

For further work, we plan to study exploiting semantic knowledge in the kernel. A promising direction is to study the different effects of these semantic similarities. We are interested in some distributional similarities (Lee, 1999) given certain context. Also the effectivenss of the semantic-enriched tree kernel in SRL is another problem.

References

Stephan Bloehdorn, Roberto Basili, Marco Cammisa, and Alessandro Moschitti. 2006. Semantic kernels for text classification based on topological measures

[4] http://l2r.cs.uiuc.edu/ cogcomp/Data/QA/QC/

[5] ftp://ftp.cs.brown.edu/pub/nlparser/

of feature similarity. In *ICDM '06: Proceedings of the Sixth International Conference on Data Mining*, pages 808–812, Washington, DC, USA. IEEE Computer Society.

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47.

X. Carreras and L. Marquez. 2005. Introduction to the conll-2005 shared task: Semantic role labeling. In *CoNLL '05: Proceedings of the 9th Conference on Computational Natural Language Learning*.

Freddy Y. Y. Choi, Peter Wiemer-hastings, and Johanna Moore. 2001. Latent semantic analysis for text segmentation. In *In Proceedings of EMNLP*, pages 109–117.

Michael Collins and Nigel Duffy. 2002. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *ACL*, pages 263–270.

Nello Cristianini, John Shawe-Taylor, and Huma Lodhi. 2002. Latent semantic kernels. *J. Intell. Inf. Syst.*, 18(2-3):127–152.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 423–429, Morristown, NJ, USA. Association for Computational Linguistics.

David Haussler. 1999. Convolution kernels on discrete structures. Technical report.

Tetsuji Kuboyama, Kilho Shin, and Hisashi Kashima. 2006. Flexible tree kernels based on counting the number of tree mappings. In *ECML/PKDD Workshop on Mining and Learning with Graphs*.

Lillian Lee. 1999. Measures of distributional similarity. In *37th Annual Meeting of the Association for Computational Linguistics*, pages 25–32.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics.

Dimitrios Mavroeidis, George Tsatsaronis, Michalis Vazirgiannis, Martin Theobald, and Gerhard Weikum. 2005. Word sense disambiguation for exploiting hierarchical thesauri in text classification. In Alípio Jorge, Luís Torgo, Pavel Brazdil, Rui Camacho, and Gama Joao, editors, *Knowledge discovery in databases: PKDD 2005 : 9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, volume 3721 of *Lecture Notes in Computer Science*, pages 181–192, Porto, Portugal. Springer.

Alessandro Moschitti, Daniele Pighin, and Roberto Basili. 2008. Tree kernels for semantic role labeling. *Comput. Linguist.*, 34(2):193–224.

Alessandro Moschitti. 2004. A study on convolution kernels for shallow semantic parsing. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 335–342, Morristown, NJ, USA. Association for Computational Linguistics.

Diarmuid Ó Séaghdha and Ann Copestake. 2008. Semantic classification with distributional kernels. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 649–656, Manchester, UK, August. Coling 2008 Organizing Committee.

Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. 2003. Using measures of semantic relatedness for word sense disambiguation. In *In Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics (CICLING-03)*, pages 241–257.

Longhua Qian, Guodong Zhou, Fang Kong, Qiaoming Zhu, and Peide Qian. 2008. Exploiting constituent dependencies for tree kernel-based semantic relation extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 697–704, Manchester, UK, August. Coling 2008 Organizing Committee.

Hiroto Saigo, Jean-Philippe Vert, Nobuhisa Ueda, and Tatsuya Akutsu. 2004. Protein homology detection using string alignment kernels. *Bioinformatics*, 20(11):1682–1689.

Kilho Shin and Tetsuji Kuboyama. 2008. A generalization of haussler's convolution kernel: mapping kernel. In *ICML*, pages 944–951.

Nianwen Xue and Martha Palmer. 2004. Calibrating features for semantic role labeling. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 88–94, Barcelona, Spain, July. Association for Computational Linguistics.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *J. Mach. Learn. Res.*, 3:1083–1106.

Min Zhang, Wanxiang Che, Aiti Aw, Chew Lim Tan, Guodong Zhou, Ting Liu, and Sheng Li. 2007. A grammar-driven convolution tree kernel for semantic role classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 200–207, Prague, Czech Republic, June. Association for Computational Linguistics.

BagPack: A general framework to represent semantic relations

Amaç Herdağdelen
CIMEC, University of Trento
Rovereto, Italy
amac@herdagdelen.com

Marco Baroni
CIMEC, University of Trento
Rovereto, Italy
marco.baroni@unitn.it

Abstract

We introduce a way to represent word pairs instantiating arbitrary semantic relations that keeps track of the contexts in which the words in the pair occur both together and independently. The resulting features are of sufficient generality to allow us, with the help of a standard supervised machine learning algorithm, to tackle a variety of unrelated semantic tasks with good results and almost no task-specific tailoring.

1 Introduction

Co-occurrence statistics extracted from corpora lead to good performance on a wide range of tasks that involve the identification of the semantic relation between two words or concepts (Sahlgren, 2006; Turney, 2006). However, the difficulty of such tasks and the fact that they are apparently unrelated has led to the development of largely *ad-hoc* solutions, tuned to specific challenges. For many practical applications, this is a drawback: Given the large number of semantic relations that might be relevant to one or the other task, we need a multi-purpose approach that, given an appropriate representation and training examples instantiating an arbitrary target relation, can automatically mine new pairs characterized by the same relation. Building on a recent proposal in this direction by Turney (2008), we propose a generic method of this sort, and we test it on a set of unrelated tasks, reporting good performance across the board with very little task-specific tweaking.

There has been much previous work on corpus-based models to extract broad classes of related words. The literature on word space models (Sahlgren, 2006) has focused on taxonomic similarity (synonyms, antonyms, co-hyponyms...) and general association (e.g., finding topically related words), exploiting the idea that taxonomically or associated words will tend to occur in similar contexts, and thus share a vector of co-occurring words. The literature on relational similarity, on the other hand, has focused on *pairs* of words, devising various methods to compare how similar the contexts in which target pairs appear are to the contexts of other pairs that instantiate a relation of interest (Turney, 2006; Pantel and Pennacchiotti, 2006). Beyond these domains, purely corpus-based methods play an increasingly important role in modeling constraints on composition of words, in particular verbal selectional preferences – finding out that, say, children are more likely to eat than apples, whereas the latter are more likely to be eaten (Erk, 2007; Padó et al., 2007). Tasks of this sort differ from relation extraction in that we need to capture *productive* patterns: we want to find out that shabu shabu (a Japanese meat dish) is eaten whereas ink is not, even if in our corpus neither noun is attested in proximity to forms of the verb *to eat*.

Turney (2008) is the first, to the best of our knowledge, to raise the issue of a unified approach. In particular, he treats synonymy and association as special cases of relational similarity: in the same way in which we might be able to tell that hands and arms are in a part-of relation by comparing the contexts in which they co-occur to the contexts of known part-of pairs, we can guess that cars and automobiles are synonyms by comparing the contexts in which they co-occur to the contexts linking known synonym pairs.

Here, we build on Turney's work, adding two main methodological innovations that allow us further generalization. First, merging classic approaches to taxonomic and relational similarity, we represent concept pairs by a vector that concatenates information about the contexts in which the two words occur independently, and the contexts in which they co-occur (Mirkin et al. 2006 also integrate information from the lexical patterns in which two words co-occur and similarity of the contexts in which each word occurs on its own, to improve performance in lexical entailment acquisition). Second, we represent contexts as bag of words and bigrams, rather than strings of words ("patterns") of arbitrary length: we leave it to the machine learning algorithm to zero in on the most interesting words/bigrams.

Thanks to the concatenated vector, we can tackle tasks in which the two words are not expected to co-occur even in very large corpora (such as selectional preference). Concatenation, together with unigram/bigram representation of context, allows us to scale down the approach to smaller training corpora (Turney used a corpus of more than 50 billion words), since we do not need to see the words directly co-occurring, and the unigram/bigram dimensions of the

Proceedings of the EACL 2009 Workshop on GEMS: GEometical Models of Natural Language Semantics, pages 33–40,
Athens, Greece, 31 March 2009.

vectors are less sparse than dimensions based on longer strings of words. We show that our method produces reasonable results also on a corpus of 2 billion words, with many unseen pairs. Moreover, our bigram and unigram representation is general enough that we do not need to extract separate statistics nor perform *ad-hoc* feature selection for each task: we build the co-occurrence matrix once, and use the same matrix in all experiments. The bag-of-words assumption also makes for faster and more compact model building, since the number of features we extract from a context is linear in the number of words in the context, whereas it is exponential for Turney. On the other hand, our method is currently lagging behind Turney's in terms of performance, suggesting that at least some task-specific tuning will be necessary.

Following Turney, we focus on devising a suitably general featural representation, and we see the specific machine learning algorithm employed to perform the various tasks as a parameter. Here, we use Support Vector Machines since they are a particularly effective general-purpose method. In terms of empirical evaluation of the model, besides experimenting with the "classic" SAT and TOEFL datasets, we show how our algorithm can tackle the selectional preference task proposed in Padó (2007) – a regression task – and we introduce to the corpus-based semantics community a challenge from the ConceptNet repository of commonsense knowledge (extending such repository by automated means is the original motivation of our project).

In the next section, we will present our proposed method along with the corpora and model parameter choices used in the implementation. In Section 3, we describe the tasks that we use to evaluate the model. Results are reported in Section 4 and we conclude in Section 5, with a brief overview of the contributions of this paper.

2 Methodology

2.1 Model

The central idea in BagPack (**Bag**-of-words representation of **Pa**ired **c**oncept **k**nowledge) is to construct a vector-based representation of a pair of words in such a way that the vector represents both the contexts where the two words co-occur and the contexts where the single words occur on their own. A straightforward approach is to construct three different sub-vectors, one for the first word, one for the second word, and one for the co-occurring pair. The concatenation of these three sub-vectors is the final vector that represents the pair.

This approach provides us a graceful fall back mechanism in case of data scarcity. Even if the two words are not observed co-occurring in the corpus – no syntagmaic information about the pair –, the corresponding vector will still represent the individual contexts where the words are observed on their own. Our hypothesis (and hope) is that this information will be representative of the semantic relation between the pair, in the sense that, given pairs characterized by same relation, there should be paradigmatic similarity across the first, resp. second elements of the pairs (e.g., if the relation is between professionals and the typical tool of their trade, it is reasonable to expect that that both professionals and tools will tend to share similar contexts).

Before going into further details, we need to describe what a "co-occurrence" precisely means, define the notion of context, and determine how to structure our vector. For a single word W, the following pseudo regular expression identifies an observation of *occurrence*:

$$\text{“}C\ W\ D\text{”} \tag{1}$$

where C and D can be empty strings or concatenations of up to 4 words separated by whitespace (i.e. $C_1, \ldots, C_i$ and $D_1, \ldots, D_j$ where $i, j \leq 4$). Each observation of this pattern constitutes a *single context* of W. The pattern is matched with the longest possible substring without crossing sentence boundaries.

Let (W_1, W_2) denote an ordered pair of words W_1 and W_2. We say the two words *occur as a pair* whenever one of the following pseudo regular expressions is observed in the corpus:

$$\text{“}C\ W_1\ D\ W_2\ E\text{”} \tag{2}$$

$$\text{“}C\ W_2\ D\ W_1\ E\text{”} \tag{3}$$

where C and E can be empty strings or concatenations of up to 2 words and similarly, D can be either an empty string or concatenation of up to 5 words (i.e. $C_1, \ldots, C_i$, $D_1, \ldots, D_j$, and $E_1, \ldots, E_k$ where $i, j \leq 2$ and $k \leq 5$). Together, patterns 2 and 3 constitute the *pair context* for W_1 and W_2. The pattern is matched with the longest possible substring while making sure that D does not contain neither W_1 nor W_2.

The number of context words allowed before, after, and between the targets are actually model parameters but for the experiments reported in this study, we used the aforementioned values with no attempt at tuning.

The vector representing (W_1, W_2) is a concatenation $\mathbf{v_1 v_2 v_{1,2}}$, where, the sub-vectors $\mathbf{v_1}$ and $\mathbf{v_2}$ are constructed by using the single contexts of W_1 and W_2 correspondingly (i.e. by pattern 1) and the sub-vector $\mathbf{v_{1,2}}$ is built by using the pair contexts identified by the patterns 2 and 3. We refer to the components as *single-occurrence vectors* and *pair-occurrence vector* respectively.

The population of BagPack starts by identifying the b most frequent unigrams and the b most frequent bigrams as *basis terms*. Let T denote a basis term. For the construction of $\mathbf{v_1}$, we create two features for each term T: t_{pre} corresponds to the number of observations of T in the single contexts of W_1 occurring before W_1 and t_{post} corresponds to the number of observations of T in the single occurrence of W_1 where T occurs after W_1 (i.e. number of observations of the pattern 1 where $T \in C$ and $T \in D$ correspondingly). The construction of $\mathbf{v_2}$ is identical except that this time the features

correspond to the number of times the basis term is observed before and after the target word W_2 in single contexts. The construction of the pair-occurrence subvector $\mathbf{v_{1,2}}$ proceeds in a similar fashion but in addition, we incorporate also the order of W_1 and W_2 as they co-occur in the pair context: The number of observations of the pair contexts where W_1 occurs before W_2 and T precedes (follows) the pair, are represented by feature t_{+pre} (t_{+post}). The number of cases where the basis term is in between the target words is represented by t_{+betw}. The number of cases where W_2 occurs before W_1 and T precedes the pair is represented by the feature t_{-pre}. Similarly the number of cases where T follows (is in between) the pair is represented by the feature t_{-post} (t_{-betw}).

Assume that the words "only" and "that" are our basis terms and consider the following context for the word pair ("cat", "lion"): "Lion is the only cat that lives in large social groups." The observation of the basis terms should contribute to the pair-occurrence subvector $\mathbf{v_{1,2}}$ and since the target words occur in reverse order, this context results in the incrementation of the features $only_{-betw}$ and $that_{-post}$ by one.

To sum up, we have $2b$ basis terms (b unigrams and b bigrams). Each of the single-occurrence sub-vectors $\mathbf{v_1}$ and $\mathbf{v_2}$ consists of $4b$ features: Each basis term gives rise to 2 features incorporating the relative position of basis term with respect to the single word. The pair-occurrence sub-vector, $\mathbf{v_{1,2}}$, consists of $12b$ features: Each basis term gives rise to 6 new features; $\times 3$ for possible relative positions of the basis term with respect to the pair and $\times 2$ for the order of the words. Importantly, the $2b$ basis terms are picked only once, and the overall co-occurrence matrix is built once and for all for *all* the tasks: unlike Turney, we do not need to go back to the corpus to pick basis terms and collect separate statistics for different tasks.

The specifics of the adaptation to each task will be detailed in Section 3. For the moment, it should suffice to note that the vectors $\mathbf{v_1}$ and $\mathbf{v_2}$ represent the contexts in which the two words occur on their own, thus encode paradigmatic information. However, $\mathbf{v_{1,2}}$ represents the contexts in which the two words co-occur, thus encode sytagmatic information.

The model training and evaluation is done in a 10-fold cross-validation setting whenever applicable. The reported performance measures are the averages over all folds and the confidence intervals are calculated by using the distribution of fold-specific results. The only exception to this setting is the SAT analogy questions task simply because we consider each question as a separate mini dataset as described in Section 3.

2.2 Source Corpora

We carried out our tests on two different corpora: ukWaC, a Web-derived, POS-tagged and lemmatized collection of about 2 billion tokens,[1] and the Yahoo! database queried via the BOSS service.[2] We will refer to these corpora as ukWaC and Yahoo from now on.

In ukWaC, we limited the number of occurrence and co-occurrence queries to the first 5000 observations for computational efficiency. Since we collect corpus statistics at the lemma level, we construct Yahoo! queries using disjunctions of inflected forms that were automatically generated with the NodeBox Linguistics library.[3] For example, the query to look for "lion" and "cat" with 4 words in the middle is: "(lion OR lions) * * * * (cat OR cats OR catting OR catted)". Each pair requires 14 Yahoo! queries (one for W_1, one for W_2, 6 for (W_1, W_2), in that order, with 0-to-5 intervening words, 6 analogous queries for (W_2, W_1)). Yahoo! returns maximally 1,000 snippets per query, and the latter are lemmatized with the TreeTagger[4] before feature extraction.

2.3 Model implementation

We did not carry out a search for "good" parameter values. Instead, the model parameters are generally picked at convenience to ease memory requirements and computational efficiency. For instance, in all experiments, b is set to 1500 unless noted otherwise in order to fit the vectors of all pairs at our hand into the computer memory.

Once we construct the vectors for a set of word pairs, we get a *co-occurrence matrix* with pairs on the rows and the features on the columns. In all of our experiments, the same normalization method and classification algorithm is used with the default parameters: First, a TF-IDF feature weighting is applied to the co-occurrence matrix (Salton and Buckley, 1988). Then following the suggestion of Hsu and Chang (2003), each feature t's $[\hat{\mu}_t - 2\hat{\sigma}_t, \hat{\mu}_t + 2\hat{\sigma}_t]$ interval is scaled to $[0, 1]$, trimming the exceeding values from upper and lower bounds (the symbols $\hat{\mu}_t$ and $\hat{\sigma}_t$ denote the average and standard deviation of the feature values respectively). For the classification algorithm, we use the C-SVM classifier and for regression the ϵ-SVM regressor, both implemented in the Matlab toolbox of Canu et al. (2005). We employed a linear kernel. The cost parameter C is set to 1 for all experiments; for the regressor, $\epsilon = 0.2$. For other pattern recognition related coding (e.g., cross validation, scaling, etc.) we made use of the Matlab PRTools (Duin, 2001).

For each task that will be defined in the next section, we evaluated our algorithm on the following representations: 1) Single-occurrence vectors ($\mathbf{v_1 v_2}$ condition) 2) Pair-occurrence vectors ($\mathbf{v_{1,2}}$ condition) 3) Entire co-occurrence matrix ($\mathbf{v_1 v_2 v_{1,2}}$ condition).

[1] http://wacky.sslmit.unibo.it

[2] http://developer.yahoo.com/search/boss/

[3] http://nodebox.net/code/index.php/Linguistics

[4] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

3 Tasks

3.1 SAT Analogy Questions

The first task we evaluated our algorithm on is the SAT analogy questions task introduced by Turney et al. (2003). In this task, there are 374 multiple choice questions with a pair of related words like (lion,cat) as the *stem* and 5 other pairs as the *choices*. The correct answer is the choice pair which has the relationship most similar to that in the stem pair.

We adopt a similar approach to the one used in Turney (2008) and consider each question as a separate binary classification problem with one positive training instance and 5 unknown pairs. For a question, we pick a pair at random from the stems of other questions as a pseudo negative instance and train our classifier on this two-instance training set. Then the trained classifier is evaluated on the choice pairs and the pair with the highest posterior probability for the positive class is called the *winner*. The procedure is repeated 10 times picking a different pseudo-negative instance each time and the choice pair which is selected as the winner most often is taken as the answer to that question. The performance measure on this task is defined as the percentage of correctly answered questions. The mean score and confidence intervals are calculated over the performance scores obtained for all folds.

3.2 TOEFL Synonym Questions

This task, introduced by Landauer and Dumais (1997), consists of 80 multiple choice questions in which a word is given as the stem and the correct choice is the word which has the closest meaning to that of the stem, among 4 candidates. To fit the task into our framework, we pair each choice with the stem word and obtain 4 word pairs for each question. The word pair constructed with the stem and the correct choice is labeled as positive and the other pairs are labeled as negative. We consider all 320 pairs constructed for all 80 questions as our dataset. Thus, the problem is turned into a binary classification problem where the task is to discriminate the synonymous word pairs (i.e. positive class) from the other pairs (i.e. negative class). We made sure that the pairs constructed for the same question were never split between training and test set, so that no question-specific learning is performed. The reason for this precaution is that the evaluation is done on a per-question basis. The estimated posterior class probabilities of the pairs constructed for the same question are compared to each other and the pair with the highest probability for the positive class is selected as the answer for the question. By keeping the pairs of a question in the same set we make sure their posteriors are calculated by the same trained classifier. The performance measure is the percentage of correctly answered questions and we report the mean performance over all 10 folds.

3.3 Selectional Preference Judgments

Linguists have long been interested in the semantic constraints that verbs impose on their arguments, a broad area that has also attracted computational modeling, with increasing interest in purely corpus-based methods (Erk, 2007; Padó et al., 2007). This task is of particular interest to us as an example of a broader class of linguistic problems that involve *productive* constraints on composition. As has been stressed at least since Chomsky's early work (Chomsky, 1957), no matter how large a corpus is, if a phenomenon is productive there will always be new well-formed instances that are not in the corpus. In the domain of selectional restrictions this is particularly obvious: we would not say that an algorithm learned the constraints on the possible objects/patients of eating simply by producing the list of all the attested objects of this verb in a very large corpus; the interesting issue is whether the algorithm can detect if an unseen object is or is not a plausible "eatee", like humans do without problems. Specifically, we test selectional preferences on the dataset constructed by Padó (2007), that collects average plausibility judgments (from 20 speakers) for nouns as either subjects or objects of verbs (211 noun-verb pairs).

We formulate this task as a regression problem. We train the ϵ-SVM regressor with 18-fold cross validation: Since the pair instances are not independent but grouped according to the verbs, one fold is constructed for each of the 18 verbs used in the dataset. In each fold, all instances sharing the corresponding verb are left out as the test set. The performance measure for this task is the Spearman correlation between the human judgments and our algorithm's estimates. There are two possible ways to calculate this measure. One is to get the overall correlation between the human judgments and our estimates obtained by concatenating the output of each cross-validation fold. That measure allows us to compare our method with the previously reported results. However, it cannot control for a possible verb-effect on the human judgment values: If the average judgment values of the pairs associated with a specific verb is significantly higher (or lower) than the average of the pairs associated with another verb, then any regressor which simply learns to assign the average value to all pairs associated with that verb (regardless of whether there is a patient or agent relation between the pairs) will still get a reasonably high correlation because of the variation of judgment scores across the verbs. To control for this effect, we also calculated the correlation between the human judgments and our estimates for each verb's plausibility values separately, and we report averages across these separate correlations (the "mean" results reported below).

3.4 Common-sense Relations from ConceptNet

Open Mind Common Sense[5] is an ongoing project of acquisition of common-sense knowledge from ordinary

[5] http://commons.media.mit.edu/en/

Relation	Pairs	Relation	Pairs
IsA	316	PartOf	139
UsedFor	198	LocationOf	1379
CapableOf	228	**Total**	1943

Table 1: ConceptNet relations after filtering.

people by letting them carry out simple semantic and linguistics tasks. An end result of the project is ConceptNet 3, a large scale semantic network consisting of relations between concept pairs (Havasi et al., 2007). It is possible to view this network as a collection of semantic assertions, each of which can be represented by a triple involving two concepts and a relation between them, e.g. *UsedFor(piccolo, make music)*. One motivation for this project is the fact that common-sense knowledge is assumed to be known by both parties in a communication setting and usually is not expressed explicitly. Thus, corpus-based approaches may have serious difficulties in capturing these relations (Havasi et al., 2007), but there are reasons to believe that they could still be useful: Eslick (2006) uses the assertions of ConceptNet as seeds to parse Web search results and augment ConceptNet by new candidate relations.

We use the ConceptNet snapshot released in June 2008, containing more than 200.000 assertions with around 20 semantic relations like UsedFor, DesiriousEffectOf, or SubEventOf. Each assertion has a confidence rating based on the number of people who expressed or confirmed that assertion. For simplicity we limited ourselves to single word concepts and the relations between them. Furthermore, we eliminated the assertions with a confidence score lower than 3 in an attempt to increase the "quality" of the assertions and focused on the most populated 5 relations of the remaining set, as given in Table 3.4. There may be more than one relation between a pair of concepts, so the total number is less than the sum of the size of the individual relation sets.

4 Results

For the multiple choice question tasks (i.e. SAT and TOEFL), we say a question is *complete* when all of the related pairs (stem and choice) are represented by vectors with at least one non-zero component. If a question has at least one pair represented by a zero-vector (*missing pairs*), then we say that the question is *partial*. For these tasks, we report the worst-case performance scores where we assume that a random guessing performance is obtained on the partial questions. This is a strict lower bound because it discards all information we have about a partial question even if it has only one missing pair. We define *coverage* as the percentage of complete questions.

4.1 SAT

In Yahoo, the coverage is quite high. In the $\mathbf{v_{1,2}}$ only condition, 4 questions had at least some choice/stem pairs with all zero components. In all other cases, all of the pairs were represented by vectors with at least one non-zero component. The highest score is obtained for the $\mathbf{v_1 v_2 v_{1,2}}$ condition with a 44.1% of correct questions, that is not significantly above the 42.5% performance of $\mathbf{v_{1,2}}$ (paired t-test, $\alpha = 0.05$). The $\mathbf{v_1 v_2}$ only condition results in a poorer performance of 33.9% correct questions, statistically lower than the former two conditions.

For ukWaC, the $\mathbf{v_{1,2}}$ only condition provides a relatively low coverage. Only 238 questions out of 374 were complete. For the other conditions, we get a complete coverage. The performances are statistically indistinguishable from each other and are 38.0%, 38.2%, and 39.6% for $\mathbf{v_{1,2}}$, $\mathbf{v_1 v_2}$, and $\mathbf{v_1 v_2 v_{1,2}}$ respectively.

Condition	Yahoo	ukWaC
$\mathbf{v_{1,2}}$	42.5%	38.0%
$\mathbf{v_1 v_2}$	33.9%	38.2%
$\mathbf{v_1 v_2 v_{1,2}}$	**44.1%**	**39.6%**

Table 2: Percentage of correctly answered questions in SAT analogy task, worst-case scenario.

In Fig. 1, the best performances we get for Yahoo and ukWaC are compared to previous studies with 95% binomial confidence intervals plotted. The reported values are taken from the ACL wiki page on the state of the art for SAT analogy questions[6]. The algorithm proposed by Turney (2008) is labeled as Turney-PairClass.

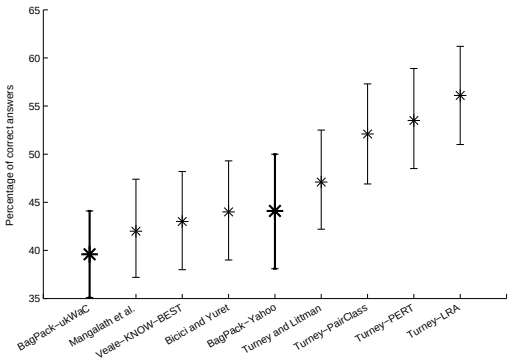


Figure 1: Comparison with previous algorithms on SAT analogy questions.

Overall, the performance of BagPack is not at the level of the state of the art but still provides a reasonable level even in the $\mathbf{v_1 v_2}$ only condition for which we do not utilize the contexts where the two words co-occur. This aspect is most striking for ukWaC where the coverage is low and by only utilizing the single-occurrence sub-vectors we obtain a performance of 38.2% correct answers (the comparable "attributional" models re-

[6]See `http://aclweb.org/aclwiki/` for further information and references

ported in Turney, 2006, have an average performance of 31%).

4.2 TOEFL

For the $\mathbf{v_{1,2}}$ sub-vector calculated for Yahoo, we have two partial questions out of 80 and the system answers 80.0% of the questions correctly. The single occurrence case $\mathbf{v_1 v_2}$ instead provides a correct percentage of 41.2% which is significantly above the random performance of 25% but still very poor. The combined case $\mathbf{v_1 v_2 v_{1,2}}$ provides a score of 75.0% with no statistically significant difference from the $\mathbf{v_{1,2}}$ case. The reason of the low performance for $\mathbf{v_1 v_2}$ is an open question.

For ukWaC, the coverage for the $\mathbf{v_1 v_2}$ case is pretty low. Out of 320 pairs, 70 were represented by zero-vectors, resulting in 34 partial questions out of 80. The performance is at 33.8%. The $\mathbf{v_1 v_2}$ case on its own does not lead to a performance better than random guessing (27.5%) but the combined case $\mathbf{v_1 v_2 v_{1,2}}$ provides the highest ukWaC score of 42.5%.

Condition	Yahoo	ukWaC
$\mathbf{v_{1,2}}$	**80.0%**	33.8%
$\mathbf{v_1 v_2}$	41.2%	27.5%
$\mathbf{v_1 v_2 v_{1,2}}$	75.0%	**42.5%**

Table 3: Percentage of correctly answered questions in TOEFL synonym task, worst-case scenario.

To our knowledge, the best performance with a purely corpus-based approach is that of Rapp (2003) who obtained a score of 92.5% with SVD. Fig. 2 reports our results and a list of other corpus-based systems which achieve scores higher than 70%, along with 95% confidence interval values. The results are taken from the ACL wiki page on the state of the art for TOEFL synonym questions.

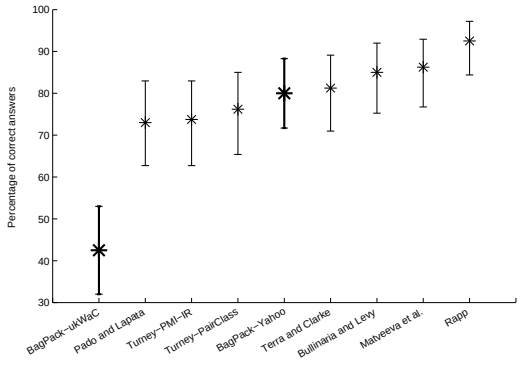


Figure 2: Comparison with previous algorithms on TOEFL synonym questions with 95% confidence intervals.

We note that our results obtained for Yahoo are comparable to the results of Turney but even the best results obtained for ukWaC and the Yahoo's results for $\mathbf{v_1 v_2}$ only condition are very poor. Whether this is because of the inability of the sub-vectors to capture synonymity or because the default parameter values of SVM are not adequate is an open question. Notice that our concatenated $\mathbf{v_1 v_2}$ vector does not exploit information about the similarity of $\mathbf{v_1}$ to $\mathbf{v_2}$, that, presumably, should be of great help in solving the synonym task.

4.3 Selectional Preference

The coverage for this dataset is quite high. All pairs were represented by non-zero vectors for Yahoo while only two pairs had zero-vectors for ukWaC. The two pairs are discarded in our experiments. For Yahoo, the best results are obtained for the $\mathbf{v_{1,2}}$ case. The single-occurrence case, $\mathbf{v_1 v_2}$, provides an overall correlation of 0.36 and mean correlation of 0.26. However low, in case of rarely co-occurring word pairs this data could be the only data we have in our hands and it is important that it provides reasonable judgment estimates.

For the ukWaC corpus, the best results we get are an overall correlation of 0.60 and a mean correlation of 0.52 for the combined case $\mathbf{v_1 v_2 v_{1,2}}$. The results for $\mathbf{v_{1,2}}$ and $\mathbf{v_1 v_2 v_{1,2}}$ are statistically indistinguishable.

	Yahoo		ukWaC	
Condition	Overall	Mean	Overall	Mean
$\mathbf{v_{1,2}}$	**0.60**	**0.45**	0.58	0.48
$\mathbf{v_1 v_2}$	0.36	0.26	0.33	0.22
$\mathbf{v_1 v_2 v_{1,2}}$	0.55	0.42	**0.60**	**0.52**

Table 4: Spearman correlations between the targets and estimations for selectional preference task.

In Fig. 3, we present a comparison of our results with some previous studies reported in Padó et al. (2007). The best result reported so far is a correlation of 0.52. Our results for Yahoo and ukWaC are currently the highest correlation values reported. Even the verb-effect-controlled correlations achieve competitive performance.

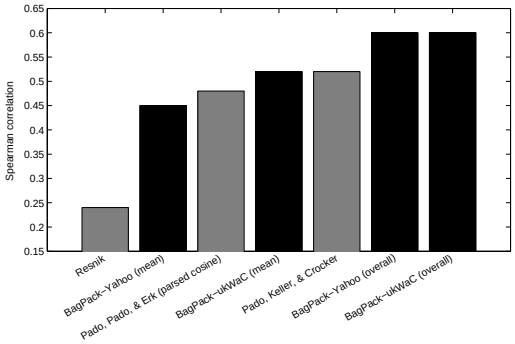


Figure 3: Comparison of algorithms on selectional preference task.

4.4 ConceptNet

Only for this task, (because of practical memory limitations) we reduced the model parameter b to 500, which means we used the 500 most frequent unigrams and 500 most frequent bigrams as our basis terms. For each of the 5 relations at our hand, we trained a different SVM classifier by labeling the pairs with the corresponding relation as positive and the rest as negative. To eliminate the issue of unbalanced number of negative and positive instances we randomly down-sampled the positive or negative instances set (whichever is larger). For the *IsA*, *UsedFor*, *CapableOf*, and *PartOf* relations, the down-sampling procedure means keeping some of the negative instances out of the training and test sets while for the *LocationOf* relation it means keeping a subset of the positive instances out. We performed 5 iterations of the down-sampling procedure and for each iteration we carried out a 10-fold cross-validation to train and test our classifier. The results are test set averages over all iterations and folds. The performance measure we use is the area under the *receiver operating characteristic* (AUC in short for area under the curve). The AUC of a classifier is the area under the curve defined by the corresponding true positive rate and false positive rate values obtained for varying the threshold of the classifier to accept an instance as positive. Intuitively, AUC is the probability that a randomly picked positive instance's estimated posterior probability is higher than a randomly picked negative instance's estimated posterior probability (Fawcett, 2006).

The coverage is quite high for both corpora: Out of 1943 pairs,only 3 were represented by a zero-vector in Yahoo while in ukWaC this number is 68. For simplicity, we discarded missing pairs from our analysis. We report only the results obtained for the entire co-occurrence matrix. The results are virtually identical for the other conditions too: Both for Yahoo and ukWaC, almost all of the AUC values obtained for all relations and for all conditions are above 95%. Only the PartOf relation has AUC values above 90% (which is still a very good result).

Relation	Yahoo	ukWaC
IsA	99.0%	98.0%
UsedFor	98.2%	98.5%
CapableOf	98.9%	99.1%
PartOf	97.6%	95.0%
LocationOf	99.0%	98.8%

Table 5: AUC scores for 5 relations of ConceptNet, classifier trained for $\mathbf{v_1 v_2 v_{1,2}}$ condition.

The very high performance we observe for the ConceptNet task is surprising when compared to the moderate performance we observe for other tasks. Our extensive filtering of the assertions could have resulted in a biased dataset which might have made the job of the classifier easy while reducing its generalization capacity. To investigate this, we decided to use the pairs coming from the SAT task as a validation set.

Again, we trained an SVM classifier on the ConceptNet data for each of the 5 relations like we did previously, but this time without cross-validation (i.e. after the down-sampling, we used the entire set as the training dataset in each iteration). Then we evaluated the classifiers on the 2224 pairs of the SAT analogy task (removing pairs that were in the training data) and averaged the posterior probability reported by each SVM over each down-sampling iteration. The 5 pairs which are assigned the highest posterior probability for each relation are reported in Table 6. We have not yet quantified the performance of BagPack in this task but the preliminary results in this table are, qualitatively, exceptionally good.

5 Conclusions

We presented a general way to build a vector-based space to represent the semantic relations between word pairs and showed how that representation can be used to solve various tasks involving semantic similarity. For SAT and TOEFL, we obtained reasonable performances comparable to the state of the art. For the estimation of selective preference judgments about verb-noun pairs, we achieved state of the art performance. Perhaps more importantly, our representation format allows us to provide meaningful estimates even when the verb and noun are not observed co-occurring in the corpus – which is an obvious advantage over the models which rely on sytagmatic contexts alone and cannot provide estimates for word pairs that are not seen directly co-occurring. We also obtained very promising results for the automated augmentation of ConceptNet.

The generality of the proposed method is also reflected in the fact that we built a single feature space based on frequent basis terms and used the same features for all pairs coming from different tasks. The use of the same feature set for all pairs makes it possible to build a single database of word-pair vectors. For example, we were able to re-use the vectors constructed for SAT pairs as a validation set in the ConceptNet task. Furthermore, the results reported here are obtained for the same machine learning model (SVM) without any parameter tweaking, which renders them very strict lower bounds.

Another contribution is that the proposed method provides a way to represent the relations between words even if they are not observed co-occurring in the corpus. Employing a larger corpus can be an alternative solution for some cases but this is not always possible and some tasks, like estimating selectional preference judgments, inherently call for a method that does not exclusively depends on paired co-occurrence observations.

Finally, we introduced ConceptNet, a common-sense semantic network, to the corpus-based semantics community, both as a new challenge and as a repository we

Rank	IsA	UsedFor	PartOf	CapableOf	LocationOf
1	watch,timepiece	pencil,draw	vehicle,wheel	motorist,drive	spectator,arena
2	emerald,gem	blueprint,build	spider,leg	volatile,vaporize	water,riverbed
3	cherry,fruit	detergent,clean	keyboard,finger	concrete,harden	bovine,pasture
4	dinosaur,reptile	guard,protect	train,caboose	parasite,contribute	benediction,church
5	ostrich,bird	buttress,support	hub,wheel	immature,develop	byline,newspaper

Table 6: Top 5 SAT pairs classified as positive for ConceptNet relations, classifier trained for $\mathbf{v_1 v_2 v_{1,2}}$ condition.

can benefit from.

In future work, one of the most pressing issue we want to explore is how to better exploit the information in the single occurrence vectors: currently, we do not make any use of the *overlap* between $\mathbf{v_1}$ and $\mathbf{v_2}$. In this way, we are missing the classic intuition that taxonomically similar words tend to occur in similar contexts, and it is thus not surprising that $\mathbf{v_1 v_2}$ flunks the TOEFL. We are currently looking at ways to augment our concatenated vector with "meta-information" about vector overlap.

References

S. Canu, Y. Grandvalet, V. Guigue and A. Rakotomamonjy. 2005. *SVM and Kernel Methods Matlab Toolbox*, Perception Systèmes et Information, INSA de Rouen, Rouen, France

N. Chomsky. 1957. *Syntactic structures*. Mouton, The Hague.

R. P. W. Duin. 2001. PRTOOLD (Version 3.1.7), A Matlab toolbox for pattern recognition. Pattern Recognition Group. Delft University of Technology.

K. Erk. 2007. A simple, similarity-based model for selectional preferences. *Proceedings of ACL 2007*.

K. Erk and S. Padó. 2008. A structured vector space model for word meaning in context. *Proceedings of EMNLP 2008*.

I. Eslick. 2006. Searching for commonsense. Master's thesis, Massachusetts Institute of Technology.

T. Fawcett. 2006. An introduction to roc analysis. *Pattern Recogn. Lett.*, 27(8):861–874.

C. Havasi, R. Speer and J. Alonso. 2007. Conceptnet 3: a flexible, multilingual semantic network for common sense knowledge. In *Recent Advances in Natural Language Processing*, Borovets, Bulgaria, September.

C.-W. Hsu, C.-C Chang. 2003. *A practical guide to support vector classification*. Technical report, Department of Computer Science, National Taiwan University.

T.K. Landauer and S.T. Dumais. 1997. A solution to Plato's problem: The Latent Semantic Analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104(2): 211–240.

H. Liu and P. Singh. 2004. ConceptNet — A practical commonsense reasoning tool-kit. *BT Technology Journal*, 22(4) 211–226.

S. Mirkin, I. Dagan and M. Geffet. 2006. Integrating pattern-based and distributional similarity methods for lexical entailment acquisition. *Proceedings of COLING/ACL 2006*, 579–586.

S. Padó, S. Padó and K. Erk. 2007. Flexible, corpus-based modelling of human plausibility judgements. *Proceedings EMNLP 2007*, 400–409.

U. Padó. 2007. *The Integration of Syntax and Semantic Plausibility in a Wide-Coverage Model of Sentence Processing*. Ph.D. thesis, Saarland University.

P. Pantel and M. Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. *Proceedings of COLING/ACL 2006*, 113–120.

R. Rapp. 2003. Word sense discovery based on sense descriptor dissimilarity. *Proceedings of MT Summit IX*: 315–322.

M. Sahlgren. 2006. *The Word-space model*. Ph.D. dissertation, Stockholm University, Stockholm.

G. Salton and C. Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5): 513–523.

R. Speer, C. Havasi and H. Lieberman. 2008. Analogyspace: Reducing the dimensionality of common sense knowledge. In Dieter Fox and Carla P. Gomes, editors, *AAAI*, pages 548–553. AAAI Press.

P. Turney. 2006. Similarity of semantic relations. *Computational Linguistics*, 32(3): 379–416.

P. Turney. 2008. A uniform approach to analogies, synonyms, antonyms and associations. *Proceedings of COLING 2008*, 905–912.

Positioning for Conceptual Development using Latent Semantic Analysis

Fridolin Wild, Bernhard Hoisl
Vienna University of Economics
and Business Administration

Gaston Burek
University of Tübingen
Computational Linguistics Division

Abstract

With increasing opportunities to learn online, the problem of positioning learners in an educational network of content offers new possibilities for the utilisation of geometry-based natural language processing techniques.

In this article, the adoption of latent semantic analysis (LSA) for guiding learners in their conceptual development is investigated. We propose five new algorithmic derivations of LSA and test their validity for positioning in an experiment in order to draw back conclusions on the suitability of machine learning from previously accredited evidence. Special attention is thereby directed towards the role of distractors and the calculation of thresholds when using similarities as a proxy for assessing conceptual closeness.

Results indicate that learning improves positioning. Distractors are of low value and seem to be replaceable by generic noise to improve threshold calculation. Furthermore, new ways to flexibly calculate thresholds could be identified.

1 Introduction

The path to new content-rich competencies is paved by the acquisition of new and the reorganisation of already known concepts. Learners willing to take this journey, however, are imposed with the problem of positioning themselves to that point in a learning network of content, where they leave their known trails and step into the unknown – and to receive guidance in subsequent further conceptual development.

More precisely, positioning requires to map characteristics from a learner's individual epistemic history (including both achievements and shortcomings) to the characteristics of the available learning materials and to recommend remedial action on how to achieve selected conceptual development goals (Van Bruggen et al., 2006).

The conceptual starting points of learners necessary to guide the positioning process is reflected in the texts they are writing. Through structure and word choice, most notably the application of professional language, arrangement and meaning of these texts give cues about the level of competency[1] development.

As learning activities increasingly leave digital traces as evidence for prior learning, positioning support systems can be built that reduce this problem to developing efficient and effective matchmaking procedures.

Latent semantic analysis (LSA) (Deerwester et al., 1990) as one technology in the family of geometry-based natural language models could in principle provide a technological basis for the positioning aims outlined above. The assumption underlying this is that the similarity to and of learning materials can be used as a proxy for similarity in learning outcomes, i.e. the developmental change in conceptual coverage and organisation caused by learning.

In particular, LSA utilises threshold values for the involved semantic similarity judgements. Traditionally the threshold is obtained by calculating the average similarity between texts that correspond to the same category. This procedure can be inaccurate if a representative set of documents for each category is not available. Furthermore, similarity values tend to decrease with increasing corpora and vocabulary sizes. Also, the role of distractors in this context, i.e. negative evidence as reference material to sharpen classification for positioning, is largely unknown.

With the following experiment, we intend to

[1] See (Smith, 1996) for a clarification of the difference of competence and competency

Proceedings of the EACL 2009 Workshop on GEMS: GEometical Models of Natural Language Semantics, pages 41–48,
Athens, Greece, 31 March 2009.

validate that geometrical models (particularly latent semantic analysis) can produce near human results regarding their propositions on how to account written learner evidence for prior learning and positioning these learners to where the best-suiting starting points are. We will show that latent semantic analysis works for positioning and that it can provide effective positioning.

The main focus of this contribution is to investigate whether machine learning proves useful for the positioning classifiers, whether distractors improve results, and what the role of thresholds for the classifiers is.

The rest of this paper is structured as follows. At first, positioning with LSA and related work are explained. This is followed by an outline of our own approach to positioning. Subsequently, a validation experiment for the set of new algorithms is outlined with which new light is shed on the utilisation of LSA for positioning. The results of this experiment are analysed in the following section in oder to, finally, yield conclusions and an outlook.

2 Positioning with LSA

According to (Kalz et al., 2007), positioning "is a process that assists learners in finding a starting point and an efficient route through the [learning] network that will foster competence building". Often, the framework within which this competence development takes places is a formal curriculum offered by an educational provider.

Not only when considering a lifelong learner, for whom the borders between formal and informal learning are absolutely permeable, recognition of prior learning turns out to be crucial for positioning: each individual background differs and prior learning needs to be respected or even accredited before taking up new learning activities – especially before enrolling in a curriculum.

Typically, the necessary evidence of prior learning (i.e., traces of activities and their outcomes) are gathered in a learner's portfolio. This portfolio is then analysed to identify both starting points and a first navigation path by mapping evidence onto the development plans available within the learning network.

The educational background represented in the portfolio can be of formal nature (e.g. certified exams) in which case standard admission and exemption procedures may apply. In other cases such standard procedures are not available, therefore assessors need to intellectually evaluate learner knowledge on specific topics. In procedures for accreditation of prior learning (APL), assessors decide whether evidence brought forward may lead to exemptions from one or more courses.

For supporting the positioning process (as e.g. needed for APL) with technology, three different computational classes of approaches can be distinguished: mapping procedures based on the analysis of informal descriptions with textmining technologies, meta-data based positioning, and positioning based on ontology mappings (Kalz et al., 2007). Latent semantic analysis is one of many possible techniques that can be facilitated to support or even partially automate the analysis of informal portfolios.

2.1 LSA

LSA is an algorithm applied to approximate the meaning of texts, thereby exposing semantic structure to computation. LSA combines the classical vector-space model with a singular value decomposition (SVD), a two-mode factor analysis. Thus, bag-of-words representations of texts can be mapped into a modified vector space that is assumed to reflect semantic structure.

The basic idea behind LSA is that the collocation of terms of a given document-term-space reflects a higher-order – latent semantic – structure, which is obscured by word usage (e.g. by synonyms or ambiguities). By using conceptual indices that are derived statistically via a truncated SVD, this variability problem is believed to be overcome.

In a typical LSA process, first a document-term matrix is constructed from a given text base of n documents containing m terms. This matrix M of the size $m \times n$ is then resolved by the SVD into the term vector matrix T (constituting the left singular vectors), the document vector matrix D (constituting the right singular vectors) being both orthonormal and the diagonal matrix S.

Multiplying the truncated matrices T_k, S_k and D_k results in a new matrix M_k (see Figure 1) which is the least-squares best fit approximation of M with k singular values (Berry et al., 1994).

2.2 Related Work

LSA has been widely used in learning applications such as automatic assessment of essays, provision

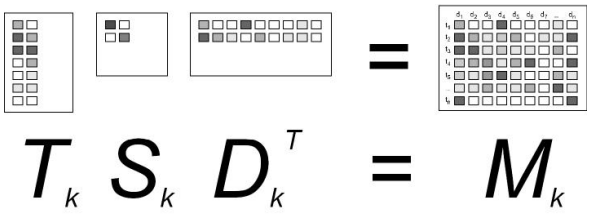


Figure 1: Reconstructing a textmatrix from the lower-order latent-semantic space.

of feedback, and selection of suitable materials according to the learner's degree of expertise in specific domains.

The Intelligent Essay Assessor (IEA) is an example of the first type of applications where the semantic space is build from materials on the topic to be evaluated. In (Foltz et al., 1999) the finding is reported that the IEA rating performance is close to the one of human raters.

In (van Bruggen et al., 2004) authors report that LSA-based positioning requires creating a latent-semantic space from text documents that model learners' and public knowledge on a specific subject. Those texts include written material of learners' own production, materials that the learner has studied and learned in the past, and descriptions of learning activities that the learner has completed in the past. Public knowledge on the specific subject includes educational materials of all kind (e.g. textbooks or articles).

In this case the description of the activity needs to be rich in the sense of terminology related to the domain of application. LSA relies on the use of rich terminology to characterize the meaning.

Following the traditional LSA procedure, the similarity (e.g. cosine) between LSA vector models of the private and public knowledge is then calculated to obtain the learner position with respect to the public knowledge.

3 Learning Algorithms for Positioning

In the following, we design an experiment, conduct it, and evaluate the results to shed new light on the use of LSA for positioning.

The basic idea of the experiment is to investigate whether LSA works for advising assessors on acceptance (or rejection) of documents presented by the learner as evidence of previous conceptual knowledge on specific subjects covered by the curriculum. The assessment is in all cases done by comparing a set of learning materials (model solutions plus previously accepted/rejected reference material) to the documents from learners' portfolios using cosines as a proxy for their semantic similarity.

In this comparison, thresholds for the cosine measure's values have to be defined above which two documents are considered to be similar. Depending on how exactly the model solutions and additional reference material are utilised, different assessment algorithms can be developed.

To validate the proposed positioning services elaborated below, we compare the automatic recommendations for each text presented as evidence with expert recommendations over the same text (external validation).

To train the thresholds and as a method for assessing the provided evidence, we propose to use the following five different unsupervised and supervised positioning rules. These configurations differ in the way how their similarity threshold is calculated and against which selection of documents (model solutions and previously expert-evaluated reference material) the 'incoming' documents are compared. We will subsequently run the experiment to investigate their effectiveness and compare the results obtained with them.

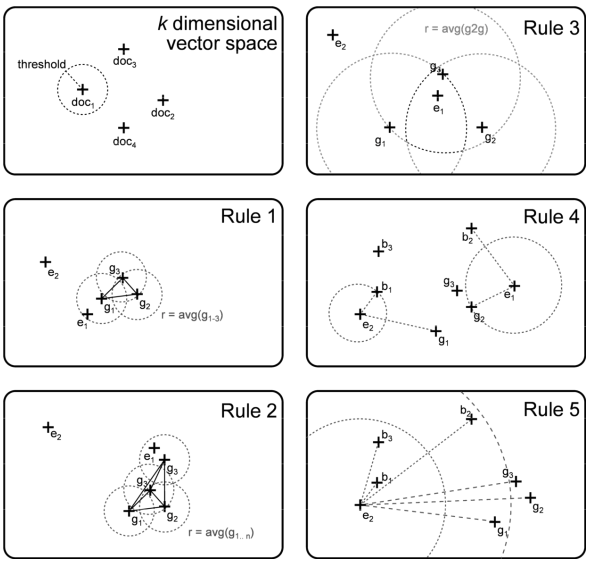


Figure 2: The five rules.

The visualisation in Figure 2 depicts the working principle of the rules described below. In each panel, a vector space is shown. Circles depict radial cosine similarity. The document representatives labelled with g_n are documents with positive evidence ('good' documents), the ones labelled with b_n are those with negative. The test docu-

ments carry the labels e_n ('essay').

Best of Golden: The threshold is computed by averaging the similarity of all three golden standard essays to each other. The similarity of the investigated essay is compared to the best three golden standard essays (=machine score). If the machine score correlates above the threshold with the human judgement, the test essay is stated correct. This rule assumes that the gold standards have some variation in the correlation among each other and that using the average correlation among the gold standards as a threshold is taking that into account.

Best of Good: Best essays of the humanly judged good ones. The assumption behind this is that with more positive examples to evaluate an investigated essay against, the precision of the evaluation should rise. The threshold is the average of the positive evidence essays among each other.

Average to Good > *Average among Good*: Tests if the similarity to the 'good' examples is higher than the average similarity of the humanly judged good ones. Assumption is that the good evidence gathered circumscribes that area in the latent semantic space which is representative of the abstract model solution and that any new essay should be within the boundaries characterised by this positive evidence thus having a higher correlation to the positive examples then they have among each other.

Best of Good > *Best of Bad*: Tests whether the maximum similarity to the good essays is higher than the maximum similarity to bad essays. If a tested essay correlates higher to the best of the good than to the best of the bad, then it is classified as accepted.

Average of Good > *average of Bad*: The same with average of good > average of bad. Assumption behind this is again that both bad and good evidence circumscribe an area and that the incoming essay is in either the one or the other class.

4 Corpus and Space Construction

The corpus for building the latent semantic space is constructed with 2/3 German language corpus (newspaper articles) and 1/3 domain-specific (a textbook split into smaller units enhanced by a collection of topic related documents which Google threw up among the first hits). The corpus has a size of 444k words (59.719 terms, 2444 textual units), the mean document length is 181 words with a standard deviation of 156. The term frequencies have a mean of 7.4 with a standard deviation of 120.

The latent semantic space is constructed over this corpus deploying the lsa package for R (Wild, 2008; Wild and Stahl, 2007) using $dimcalc\_share$ as the calculation method to estimate a good number of singular values to be kept and the standard settings of $textmatrix()$ to pre-process the raw texts. The resulting space utilises 534 dimensions.

For the experiment, 94 essays scored by a human evaluator on a scale from 0 to 4 points where used. The essays have a mean document length of 22.75 terms with a standard deviation of 12.41 (about one paragraph).

To estimate the quality of the latent semantic space, the learner writings were folded into the semantic space using $fold\_in()$. Comparing the non-partitioned (i.e. 0 to 4 in steps of .5) human scores with the machine scores (average similarity to the three initial model solutions), a highly significant trend can be seen that is far from being perfect but still only slightly below what two human raters typically show.

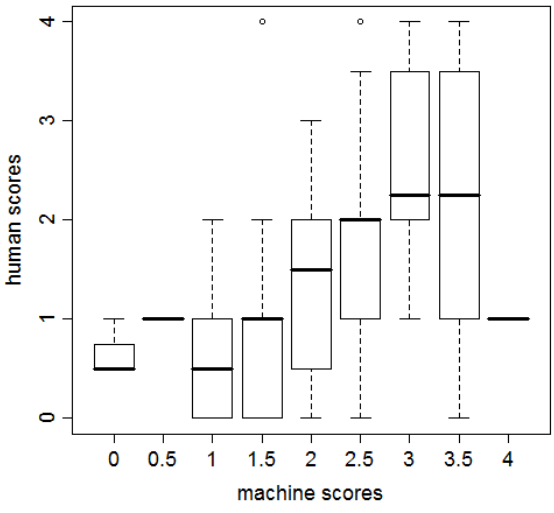


Figure 3: Human vs. Machine Scores.

Figure 3 shows the qualitative human expert judgements versus the machine grade distribution using the non-partitioned human scores (from 0 to 4 points in .5 intervals) against the rounded average cosine similarity to the initial three model solutions. These machine scores are rounded such that they – again – create the same amount of intervals. As can be seen in the figure, the extreme

of each score level is displayed in the upper and lower whisker. Additionally, the lower and upper 'hinge' and the median are shown. The overall Spearman's rank correlation of the human versus the (continuous) machine scores suggests a with .51 medium effect being highly significant on a level with the p-value below .001. Comparing this to untrained human raters, who typically correlate around .6, this is in a similar area, though the machine differences can be expected to be different in nature.

A test with 250 singular values was conducted resulting in a considerately lower Spearman correlation of non-partitioned human and machine scores.

Both background and test corpus have deliberately been chosen from a set of nine real life cases to serve as a prototypical example.

For the experiment, the essay collection was split by half into training (46) and test (48) set for the validation. Each set has been partitioned into roughly an equal number of accepted (scores < 2, 22 essays in training set, 25 in test) and rejected essays (scores $>= 2$, 24 essays in training, 23 in test). All four subsets, – test and training partitioned into accepted and rejected –, include a similarly big number of texts.

In order to cross validate, the training and test sets were random sampled ten times to get rid of influences on the algorithms from the sort order of the essays. Both test and training sets were folded into the latent semantic space. Then, random sub samples (see below) of the training set were used to train the algorithms, whereas the test set of 48 test essays in each run was deployed to measure precision, recall, and the f-measure to analyse the effectiveness of the rules proposed.

Similarity is used as a proxy within the algorithms to determine whether a student writing should be accepted for this concept or rejected. As similarity measure, the cosine similarity $cosine()$ was used.

In each randomisation loop, the share of accepted and rejected essays to learn from was varied in a second loop of seven iterations: Always half of the training set essays were used and the amount of accepted essays was decreased from 9 to 2 while the number of rejected essays was increased from 2 to 9. This way, the influence of the number of positive (and negative) examples could be investigated.

This mixture of accepted and rejected evidence to learn from was diversified to investigate the influence of learning from changing shares and rising or decreasing numbers of positive and/or negative reference documents – as well as to analyse the influence of recalculated thresholds. While varying these training documents, the human judgements were given to the machine in order to model learning from previous human assessor acceptance and rejection.

5 Findings

5.1 Precision versus Recall

The experiments where run with the five different algorithms and with the sampling procedures described above. For each experiment precision and recall where measured to find out if an algorithm can learn from previous inputs and if it is better or worse compared to the others.

As mentioned above, the following diagrammes depict from left to right a decreasing number of accepted essays available for training (9 down to 2) while the number of rejected essays made available for training is increased (from 2 to 9).

Rule 1 to 3 do not use these negative samples, rule 1 does not even use the positive samples but just three additional model solutions not contained in the training material of the others. The curves show the average precision, recall, and f-measure[2] of the ten randomisations necessary for the cross validation. The size of the circles along the curves symbolises the share of accepted essays in the training set.

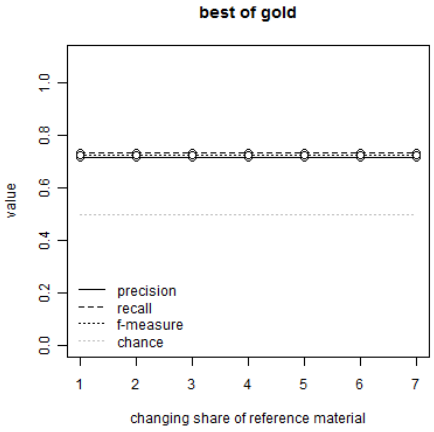


Figure 4: Rule 1: Best of Three Golden

[2] $F = 2 \cdot \frac{precision \cdot recall}{precision + recall}$

Figure 4 shows that recall and precision stay stable as there are no changes to the reference material taken into account: all essays are evaluated using three fixed 'gold standard' texts. This rule serves as a baseline benchmark for the other results.

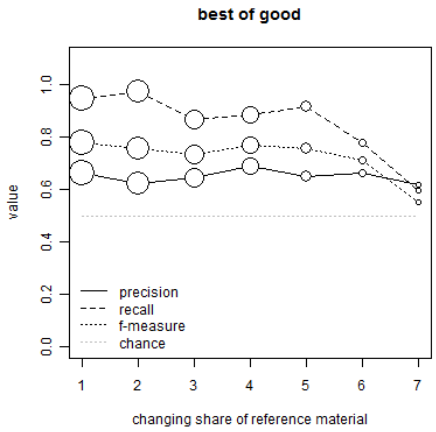


Figure 5: Rule 2: Best of Good

Figure 5 depicts a falling recall when having less positively judged essays in the training sample. In most cases, the recall is visibly higher than in the first rule, 'Best of Gold', especially when given enough good examples to learn from. Precision is rather stable. We interpret that the falling recall can be led back to the problem of too few examples that are then not able to model the target area of the latent semantic space.

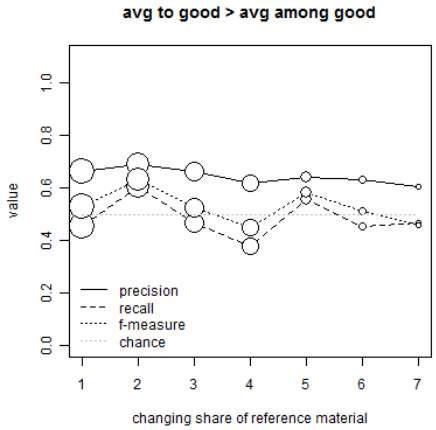


Figure 6: Rule 3: Avg of Good > Avg among Good

Figure 6 displays that the recall worsens and is very volatile[3]. Precision, however, is very stable and slightly higher than in the previous rule, especially with rising numbers of positive examples. It seems that the recall is very dependant on the positive examples whether they are able to characterise representative boundaries: seeing recall change with varying amounts of positive examples, this indicates that the boundaries are not very well chosen. We assume that this is related to containing 'just pass' essays that were scored with 2.0 or 2.5 points and distort the boundaries of the target area in the latent semantic concept space.

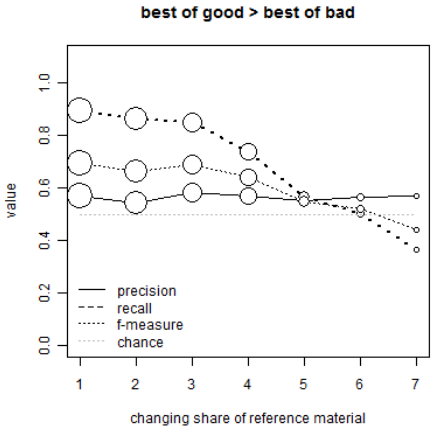


Figure 7: Rule 4: Best of Good > Best of Bad

Figure 7 exhibits a quickly falling recall, though starting on a very high level, whereas precision is relatively stable. Having more negative evidence clearly seems to be counter productive and it seems more important to have positive examples to learn from. We have two explanations for this: First, bad examples scatter across the space and it is likely for a good essay to correlate higher with a bad one when there is only a low number of positive examples. Second, bad essays might contain very few words and thus expose correlation artefacts that would in principle be easy to detect, but not with LSA.

Figure 8 depicts a recall that is generically higher than in the 'Best of Gold' case, while precision is in the same area. Recall seems not to be so stable but does not drop with more bad samples (and less good ones) to learn from such as in the 'Best of Good' case. We interpret that noise can be added to increase recall while still only a low number of positive examples is available to improve it.

[3]We analysed the recall in two more randomisations of the whole experiment; whereas the other rules showed the same results, the recall of this rule was unstable over the test runs, but in tendency lower than in the other rules.

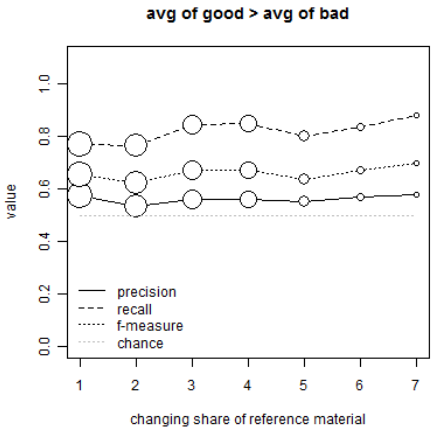


Figure 8: Rule 5: Avg of Good > Avg of Bad

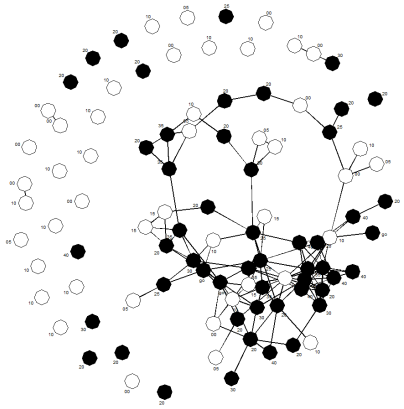


Figure 10: Network with filtered vocabulary.

5.2 Clustering

To gain further insight about the location of the 94 essays and three gold standards in the higher order latent-semantic space, a simple cluster analysis of their vectors was applied. Therefore, all document-to-document cosine similarities were calculated, filtered by a threshold of .65 to capture only strong associations, and, subsequently, a network plot of this resulting graph was visualised.

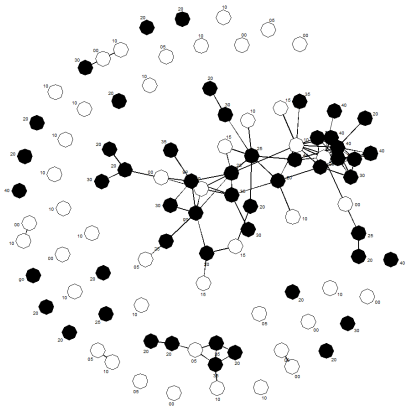


Figure 9: Similarity Network (cos >= .65).

As can be seen in the two charts, the humanly positively judged evidence seems to cluster quite well in the latent-semantic space when visualised as a network plot. Through filtering the document vectors by the vocabulary used only in the accepted, rejected, or both classes, an even clearer picture could be generated, shown in Figure 10.

Both figures clearly depict a big connected component consisting mainly out of accepted essays, whereas the rejected essays mainly spread in the unconnected surrounding. The rejected essays are in general not similar to each other, whereas the accepted samples are.

The second Figure 10 is even more homogeneous than the first due to the use of the restricted vocabulary (i.e. the terms used in all accepted and rejected essays).

6 Conclusion and Outlook

Distractors are of low value in the rules tested. It seems that generic noise can be added to keep recall higher when only a low number of positive examples can be utilised. An explanation for this can be found therein that there are always a lot more heterogeneous ways to make an error. Homogeneity can only be assumed for the positive evidence, not for negative evidence.

Noise seems to be useful for the calculation of thresholds. Though it will need further investigation whether our new hypothesis works that bad samples can be virtually anything (that is not good).

Learning helps. The recall was shown to improve in various cases, while precision stayed at the more or less same level as the simple baseline rule. Though the threshold calculation using the difference to good and bad examples seemed to bear the potential of increasing precision.

Thresholds and ways how to calculate them are evidently important. We proposed several well working ways on how to construct thresholds from evidence that extend the state of the art. Thresholds usually vary with changing corpus sizes and the measures proposed can adopt to that.

We plan to investigate the use of support vec-

tor machines in the latent semantic space in order to gain more flexible means of characterising the boundaries of the target area representing a concept.

It should be mentioned that this experiment demonstrates that conceptual development can be measured and texts and their similarity can serve as a proxy for that. Of course the experiment we have conducted bears the danger to bring results that are only stable within the topical area chosen.

We were able to demonstrate that textual representations work on a granularity level of around 23 words, i.e. with the typical length of a free text question in an exams.

While additionally using three model solutions or at least two positive samples, we were able to show that using a textbook split into paragraph-sized textual units combined with generic background material, valid classifiers can be built with relative ease. Furthermore, reference material to score against can be collected along the way.

The most prominent open problem is to try and completely get rid of model solutions as reference material and to assess the lower level concepts (terms and term aggregates) directly to further reduce corpus construction and reference material collection. Using clustering techniques, this will mean to identify useful ways for efficient visualisation and analysis.

7 Acknowledgements

This work has been financially supported by the European Union under the ICT programme of the 7th Framework Programme in the project 'Language Technology for Lifelong Learning'.

References

Michael Berry, Susain Dumais, and Gavin O'Brien. 1994. Using linear algebra for intelligent information retrieval. Technical Report CS-94-270, Department of Computer Science, University of Tennessee.

Scott Deerwester, Susan Dumais, Georg W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Peter Foltz, Darrell Laham, and Thomas K. Landauer. 1999. Automated essay scoring: Applications to educational technology. In Collis and Oliver, editors, *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 1999*, pages 939–944, Chesapeake, VA. AACE.

Marco Kalz, Jan Van Bruggen, Ellen Rusmann, Bas Giesbers, and Rob Koper. 2007. Positioning of learners in learning networks with content analysis, metadata and ontologies. *Interactive Learning Environments*, (2):191–200.

Mark K. Smith. 1996. Competence and competency. http://www.infed.org/biblio/b-comp.htm.

Jan van Bruggen, Peter Sloep, Peter van Rosmalen, Francis Brouns, Hubert Vogten, Rob Koper, and Colin Tattersall. 2004. Latent semantic analysis as a tool for learner positioning in learning networks for lifelong learning. *British Journal of Educational Technology*, (6):729–738.

Jan Van Bruggen, Ellen Rusman, Bas Giesbers, and Rob Koper. 2006. Content-based positioning in learning networks. In Kinshuk, Koper, Kommers, Kirschner, Sampson, and Didderen, editors, *Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies*, pages 366–368, Kerkrade, The Netherlands.

Fridolin Wild and Christina Stahl. 2007. Investigating unstructured texts with latent semantic analysis. In Lenz and Decker, editors, *Advances in Data Analysis*, pages 383–390, Berlin. Springer.

Fridolin Wild. 2008. lsa: Latent semantic analysis. r package version 0.61.

Semantic similarity of distractors in multiple-choice tests: extrinsic evaluation

Ruslan Mitkov, Le An Ha, Andrea Varga and **Luz Rello**
University of Wolverhampton
Wolverhampton, UK
{R.Miktov, L.A.Ha, Andrea.Varga,
L.RelloSanchez}@wlv.ac.uk

Abstract

Mitkov and Ha (2003) and Mitkov et al. (2006) offered an alternative to the lengthy and demanding activity of developing multiple-choice test items by proposing an NLP-based methodology for construction of test items from instructive texts such as textbook chapters and encyclopaedia entries. One of the interesting research questions which emerged during these projects was how better quality distractors could automatically be chosen. This paper reports the results of a study seeking to establish which similarity measures generate better quality distractors of multiple-choice tests. Similarity measures employed in the procedure of selection of distractors are collocation patterns, four different methods of WordNet-based semantic similarity (extended gloss overlap measure, Leacock and Chodorow's, Jiang and Conrath's as well as Lin's measures), distributional similarity, phonetic similarity as well as a mixed strategy combining the aforementioned measures. The evaluation results show that the methods based on Lin's measure and on the mixed strategy outperform the rest, albeit not in a statistically significant fashion.

1 Introduction

Multiple-choice tests are sets of test items, the latter consisting of a question or *stem* (e.g. Who was voted the best international footballer for 2008?), the correct *answer* (e.g. Ronaldo) and *distractors* (e.g. Messi, Ronaldino, Torres). This type of test has proved to be an efficient tool for measuring students' achievement and is used on a daily basis both for assessment and diagnostics worldwide.[1] According to Question Mark Computing Ltd (p.c.), who have licensed their Perception software to approximately three million users so far, 95% of their users employ this software to administrate multiple-choice tests.[2] Despite their popularity, the manual construction of such tests remains a time-consuming and labour-intensive task. One of the main challenges in constructing a multiple-choice test item is the selection of plausible alternatives to the correct answer which will better distinguish confident students from unconfident ones.

Mitkov and Ha (2003) and Mitkov et al. (2006) offered an alternative to the lengthy and demanding activity of developing multiple-choice test items by proposing an NLP-based methodology for construction of test items from instructive texts such as textbook chapters and encyclopaedia entries. This methodology makes use of NLP techniques including shallow parsing, term extraction, sentence transformation and semantic distance computing and employs resources such as corpora and ontologies like WordNet. More specifically, the system identifies important terms in a textbook text,

[1] This paper is not concerned with the issue of whether multiple-choice tests are better assessment methodology that other types of tests. What it focuses is on improving our new NLP methodology to generate multiple-choice tests about facts explicitly stated in single declarative sentences by establishing which semantic similarity measures give rise to better distractors.

[2] More information on the Perception software can be found at: www.questionmark.com/perception

Proceedings of the EACL 2009 Workshop on GEMS: GEometical Models of Natural Language Semantics, pages 49–56,
Athens, Greece, 31 March 2009.

transforms declarative sentences into questions and mines for terms which are semantically close to the correct answer, to serve as distractors.

The system for generation of multiple-choice tests described in Mitkov and Ha (2003) and in Mitkov et al. (2006) was evaluated in practical environment where the user was offered the option to post-edit and in general to accept, or reject the test items generated by the system[3]. The formal evaluation showed that even though a significant part of the generated test items had to be discarded, and that the majority of the items classed as 'usable' had to be revised and improved by humans, the quality of the items generated and proposed by the system was not inferior to the tests authored by humans, were more diverse in terms of topics and very importantly – their production needed 4 times less time than the manually written items. The evaluation was conducted both in terms of measuring the time needed to develop test items and in terms of classical test analysis to assess the quality of test items.

The paper is structured as follows. Section 2 will outline the importance of distractors in multiple-choice testing as the different strategies for automatic selection of the distractors are the subject of this study. Section 3 will describe how test items are produced and will detail the different strategies (semantic similarity measures and phonetic similarity) used for the selection of distractors. Section 4 outlines the in-class experiments, presents the evaluation methodology, reports on the results and discusses these results.

2 The importance of quality distractors

One of the interesting research questions which emerged during the above research was how better quality distractors could automatically be chosen. In fact user evaluation showed that from the three main tasks performed in the generation of multiple-choice tests (term identification, sentence transformation and distractor selection), it was distractor selection which needed further improvement with a view to putting it in practical use.

Distractors play a vital role for the process of multiple-choice testing in that good quality distractors ensure that the outcome of the tests provides more credible and objective picture of the knowledge of the testees involved. On the other hand, poor distractors would not contribute much to the accuracy of the assessment as obvious or too easy distractors will pose no challenge to the students and as a result, will not be able to distinguish high performing from low performing learners.

The principle according to which the distractors were chosen, was semantic similarity (Mitkov and Ha, 2003). The semantically closer were the distractors to the correct answer, the most 'plausible' they were deemed to be. The rationale behind this consists in the fact that distractors semantically distant from the correct answer could make guessing a 'straightforward task'. By way an example, if processing the sentence 'Syntax is the branch of linguistics which studies the way words are put together into sentences', the multiple-choice generation system would identify *syntax* as an important term, would transform the sentence into the question 'Which branch of linguistics studies the way words are put together into sentences?' and would choose 'Pragmatics', 'Morphology' and 'Semantics' as distractors to the correct answer 'Syntax', being closer to it than 'Chemistry', 'Football' or 'Beer' for instance (which if offered as distractors, would be easily dismissed by people who do not have even any knowledge of linguistics).

While the semantic similarity premise appears as a logical way forward to automatically select distractors, there are different methods or measures which compute semantic similarity. Each of these methods could be evaluated individually but here we evaluate their suitability for the task of selection of distractors in multiple-choice tests. This type of evaluation could be regarded as *extrinsic evaluation* of each of the methods, where the benchmark for their performance would not be an annotated corpus or human judgement on accuracy, but to what extent a specific NLP application can benefit from employing a method.

Another premise that this study seeks to verify is whether orthographically close distractors, in addition to being semantically related, could yield even better results.

[3] A post-editor's interface was developed to this end.

3 Production of test items and selection of distractors

Test items were constructed by a program based on the methodology described in the previous section. We ran the program on an on-line course materials in linguistics (Vajda, 2001). A total of 144 items were initially generated. 31 out of these 144 items were kept for further considerations as they either did not need any or, only minor revision. The remaining 113 items were deemed to require major post-editing revision. The 31 items kept for consideration were further revised by a second linguist and finally, we narrowed down the selection to 20 questions for the experiments[4]. These 20 questions gave a rise to a total of eight different assessments. Each assessment had the same 20 questions but they differed in the sets of distractors as these were chosen using different similarity measures[5] (sections 3.1-3.5).

To generate a list of distractors for single-word terms the function *coordinate terms* in WordNet is employed. For multi-word terms, noun phrases with the same head as the correct answers appearing in the source text as well as entry terms from Wikipedia having the same head with the correct answers, are used to compile the list of distractors. This list of distractors is offered to the user from which he or she could choose his/her preferred distractors.

In this study we explore which is the best way to narrow down the distractors to the 4 most suitable ones. To this end, the following strategies for computing semantic (and in one case, phonetic) similarity were employed: (i) collocation patterns, (ii-v) four different methods of WordNet-based semantic similarity (Extended gloss overlap measure, Leacock and Chodorow's, Jiang and Conrath's and Lin's measures), (vi) Distributional Similarity, and (vii) Phonetic similarity.

[4] The following is an example of an item generated of the program and then post-edited.
"Which type of clause might contain verb and dependent words? i) verb clause ii) adverb clause iii) adverbial clause
iv) multiple subordinate clause v) subordinate clause".

[5] It should be noted that there were cases where the different selection/similarity strategies picked the same distractors.

3.1 Collocation patterns

The collocation extraction strategy used in this experiment is based on the method reported in (Mitkov and Ha, 2003). Distractors that appear in the source text are given preference. If there are not enough distractors, distractors are selected randomly from the list.

For the other methods described below (sections 3.2-3.5), instead of giving preference to noun phrases appearing in the same text, and randomly pick the rest from the list, we ranked the distractors in the list based on the similarity scores between each distractor and the correct answer and chose the top 4 distractors.

We compute similarity for words rather than multi-word terms. When the correct answers and distractors are multi-word terms, we calculate the similarities between their modifier words. By way of example, in the case of "verb clause" and "adverbial clause", the similarity score between "verb" and "adverbial" is computed. When the correct answer or distractor contains more than one modifiers we compute the similarity for each modifier pairs and we choose the maximum score. (e.g. for "verb clause" and "multiple subordinate clause", similarity scores of "verb" and "multiple" and of "verb" and "subordinate" are calculated, the higher one is considered to represent the similarity score).

3.2 Four different methods for WordNet-based similarity

For computing WordNet-based semantic similarity we employed the package made available by Ted Pedersen[6]. Pedersen's tool computes (i) extended gloss overlap measure (Banerjee and Pedersen, 2003), (ii) Leacock and Chodorow's (1998) measure, (iii) Jiang and Conrath's (1997) measure and (iv) Lin's (1997) measure.

The extended gloss overlap measure calculates the overlaps between not only the definitions of the two concepts measured but also among those concepts to which they are related. The relatedness score is the sum of the squares of the overlap lengths.

Leacock and Chodorow's measure uses the normalised path length between the two concepts c_1 and c_2 and is computed as follows:

[6] http://search.cpan.org/~tpederse/WordNet-Similarity

$$sim_{lch}(c_1, c_2) = -\log\left[\frac{len(c_1, c_2)}{(2 \times MAX)}\right] \quad (1)$$

where *len* is the number of edges on the shortest path in the taxonomy between the two concepts and *MAX* is the depth of the taxonomy.

Jiang and Conrath's measure compares the sum of the information content of the individual concepts with that of their lowest common subsumer:

$$sim_{jcn}(c_1, c_2) = \frac{1}{IC(c_1) + IC(c_2) - 2 \times IC(lcs(c_1, c_2)} \quad (2)$$

where *IC(c)* is the information content (Patwardhan et al., 2003) of the concept *c*, and *lcs* denotes the lowest common subsumer, which represents the most specific concept that the two concepts have in common.

The Lin measure scales the information content of lowest common subsumer with the sum of information content of two concepts.

$$sim_{lin}(c_1, c_2) = \frac{2 \times IC(lcs(c_1, c_2))}{IC(c_1) + IC(c_2)} \quad (3)$$

3.3 Distributional similarity

For computing distributional similarity we made use of Viktor Pekar's implementation[7] based on Information Radius, which according to a comparative study by Dagan et al. (1997) performs consistently better than the other similar measures. Information Radius (or Jensen-Shannon divergence) is a variant of Kullback-Leiber divergence measuring similarity between two words as the amount of information contained in the difference between the two corresponding co-occurrence vectors. Every word w_j is presented by the set of words $w_{i1...n}$ with which it co-occurs. The semantics of w_j are modelled as a vector in an *n*-dimensional space where *n* is the number of words co-occurring with w_j, and the features of the vector are the probabilities of the co-occurrences established from their observed frequencies, as in (4). In Pekar's implementation, if one word is identified as dependent on another word by a dependency parser, these two words are said to be "co-occuring"[8]. The corpus used to collect the co-occurance vector was the BNC and the dependency parsed used the FDG parser (Tapanainen and Järvinen, 1997). The Information Radius (JS) is calculated using (5).

$$C(w_j) = \left\langle P(w_j \mid w_{i1}), P(w_j \mid w_{i2}), ..., P(w_j \mid w_{in}) \right\rangle \quad (4)$$

$$JSD(C(w_j) \,||\, C(w_k)) = \frac{1}{2} D(C(w_j) \,||\, M) + \frac{1}{2} D(C(w_k) \,||\, M) \quad (5)$$

where $M = \frac{1}{2}(C(w_j) + C(w_k))$

3.4 Phonetic similarity

For measuring phonetic similarity we use Soundex, phonetic algorithm for indexing words by sound. It operates on the principle of term based evaluation where each term is given a Soundex code. Each Soundex code itself consists of a letter and three numbers between 0 and 6. By way of example the Soundex code of *verb* is *V610* (the first character in the code is always the first letter of the word encoded). Vowels are not used and digits are based on the consonants as illustrate by the following table:

1. B, P, F, V
2. C, S, K, G, J, Q, X, Z
3. D, T
4. L
5. M, N
6. R

Table 1 Digits based on consonants

First the Soundex code for each word is generated[9]. Then similarity is computed using the Difference method, returning an integer result ranging in value from 1 (least similar) to 4 (most similar).

3.5 Mixed Strategy

After items have been generated by the above seven methods, we pick three items from each method, except from Soundex, where only two items have been picked, to compose an

[7] http://clg.wlv.ac.uk/demos/similarity/index.html

[8] There are many other ways to construct the co-occurrence vectors. This paper does not intend to exploit these different ways.

[9] We adopt the phonetic representation used in MS SQL Server. As illustrated above, each soundex code consists of a letter and three numbers, such as A252.

assessment of 20 items. This assessment is called "mixed", and used to assess whether or not an assessment with distractors generated by combining different methods would produce a different result from an assessment featuring distractors generated by a single method.

4 In-class experiments, evaluation, results and discussion

The tests (papers) generated with the help of our program with the distractors chosen according the different methods described above, were taken by a total of 243 students from different European universities: University of Wolverhampton (United Kingdom), University College Ghent (Belgium), University of Saarbrücken (Germany), University of Cordoba (Spain), University of Sofia (Bulgaria). A prerequisite for the students taking the test was that they studied language and linguistics and that they had a good command of English. Each test paper consisted of 20 questions and the students had 30 minutes to reply to the questions. The tests were offered through the Questionmark Perception web-based testing software which in addition to providing a user-friendly interface, computes diverse statistics related to the test questions answered.

In order to evaluate the quality of the multiple-choice test items generated by the program (and subsequently post-edited by humans), we employed standard *item analysis*. Item analysis is an important procedure in classical test theory which provides information as to how well each item has functioned. The item analysis for multiple-choice tests usually consists of the following information (Gronlund, 1982): (i) the difficulty of the item, (ii) the discriminating power and (iii) the usefulness[10] of each distractor. This information can tell us if a specific test item was too easy or too hard, how well it discriminated between high and low scorers on the test and whether all of the alternatives functioned as intended. Such types of analysis help improve test items or discard defective items.

Whilst this study focuses on the quality of the distractors generated, we believe that the distractors are essential for the quality of the overall test and hence the *difficulty* of an item and its *discriminating power* are deemed appropriate to assess the quality of distractors, even though the quality of the test stem also pays in important part. On the other hand usefulness is a completely independent measure as it looks at distractors only and not only the combination of stems and distractors.

In order to conduct this type of analysis, we used a simplified procedure, described in (Gronlund, 1982). We arranged the test papers in order from the highest score to the lowest score. We selected one third of the papers and called this the upper group. We also selected the same number of papers with the lowest scores and called this the lower group. For each item, we counted the number of students in the upper group who selected each alternative; we made the same count for the lower group.

(i) Item Difficulty

We estimated the *Item Difficulty* (ID) by establishing the ratio of students from the two groups who answered the item correctly (ID = C/T, where C is the number who answered the item correctly and T is the total number of students who attempted the item). As Table 2 shows, from the items featuring distractors generated using the collocation method[11], there were 4 too easy and 0 too difficult items.[12] The average Item Difficulty was 0.61. From the items with distractors generated using WordNet-based similarity[13], the results were the following. When employing the extended gloss overlap measure there were 2 too easy and 0 too difficult items, showing an average ID of 0.58. Leacock and Chodorow's measure produced 1 too easy and 3 too difficult items with item average difficulty of 0.54. The use of Jiang and Conrath's measure resulted in 3 too easy and 1 too difficult items; the average item difficulty observed was 0.57. Lin's measure delivered the best results from the

[10] Originally called 'effectiveness'. We chose to term this type of analysis 'usefulness' to distinguish it from the (cost/time) 'effectiveness' of the semi-automatic procedure as opposed to the manual construction of tests.

[11] Henceforth referred to as 'collocation items'; the distractors generated are referred to as 'collocation distractors'.

[12] For experimental purposes, we consider an item to be 'too difficult' if ID ≤ 0.15 and an item 'too easy' if ID ≥ 0.85.

[13] Henceforth referred to as 'WordNet items'; the distractors are referred to as 'WordNet distractors'.

point of item difficulty with an almost ideal average item difficulty of 0.51 (the recommended item difficult is 0.5; see also footnote 16); there were 2 too easy and 1 too difficult items.

The items constructed on the basis of distractors selected via the distributional similarity metric[14], scored an average ID of 0.64 with 6 items being too easy and 1 — too difficult. From the items with distractors produced using the phonetic similarity algorithm[15], there were 4 too easy and 0 too difficult questions with overall average difficult of 0.60. Finally, a mixed strategy produced test items with average difficulty of 0.53, 1 of them being too easy and 0 — too difficult.

The results showed that almost all items produced after selecting distractors using the strategies described above, featured very reasonable ID values. In many cases the average values were close to the recommended ID value of 0.5 with Lin's measure delivering the best ID of 0.51. Runners-up are the mixed strategy delivering items with average ID 0.53 Leacock and Chodorow's measure contributing to the generation of items with average ID of 0.54.

(ii) Discriminating Power

We estimated the item's *Discriminating Power* (DP) by comparing the number students in the upper and lower groups who answered the item correctly. It is desirable that the discrimination is *positive* which means that the item differentiates between students in the same way that the total test score does.[16] The formula for computing the *Discriminating Power* is as follows: DP = $(C_U - C_L)$: T/2, where C_U is the number of students in the upper group who answered the item correctly and C_L the number of the students in the lower group that did so. Here again T is the total number of students included in the item analysis.[17]

The average Discriminating Power for the collocation items was 0.33 and there were no negative discriminating collocation test items.[18] The figures associated to the WordNet items were as follows. The average DP for items produced with the extended gloss overlap measure was 0.32, and there were 2 items with negative discrimination. Leacock and Chodorow's measure did not produce any items with negative discrimination and the average DP of these was 0.38. Jiang and Conrath's measure gave rise to 2 negatively discriminating items and the average DP of the items based on this measure was 0.29. The selection of distractors with Lin's measure resulted in items with average DP of 0.37; none of them had a negative discrimination.

The average discrimination power for the distributional items was 0.29 (1 item with negative discrimination) and for phonetic items – 0.34 (0 item with negative discrimination). The employment of mixed strategy when selecting distractors which resulted in items with average DP of 0.39 (0 items with negative discrimination).

The figures related to the Discriminating Power of the items generated showed that whereas the DP was not of the desired high level, as a whole the proportion of items with negative discrimination was fairly low (Table 2). The items did not differ substantially in terms of the values of DP, the top performer being the items where the distractors were selected on the basis of the mixed strategy, followed by those selected by Leacock and Chodorow's measure and phonetic similarity.

(iii) Usefulness of the distractors

The *usefulness of the distractors* is estimated by comparing the number of students in the upper and lower groups who selected each incorrect alternative. A good distractor should attract more students from the lower group than the upper group.

The evaluation of the distractors estimated the average difference between students in the

[14] Henceforth referred to as 'distributional items'; the distractors are referred to as 'distributional distractors'.

[15] Henceforth referred to as 'phonetic items'; the distractors are referred to as 'phonetic distractors'.

[16] Zero DP is obtained when an equal number of students in each group respond to the item correctly. On the other hand, negative DP is obtained when more students in the lower group than the upper group answer correctly. Items with zero or negative DP should be either discarded or improved.

[17] Maximum positive DP is obtained only when all students in the upper group answer correctly and no one in the lower group does. An item that has a maximum DP (1.0) would have an ID 0.5; therefore, test authors are advised to construct items at the 0.5 level of difficulty.

[18] Obviously a negative discriminating test item is not regarded as a good one.

	Item Difficulty			Item Discriminating Power		Usefulness of distractors		
	average item difficulty	too easy	too difficult	average discriminating power	negative discriminating power	poor	not useful	average difference
Collocation items	0.61	4	0	0.33	0	2	24	0.74
WordNet items								
- Extended gloss overlap	0.58	2	0	0.32	2	9	17	0.71
- Leacock and Chodorow	0.54	1	3	0.38	0	9	20	0.76
- Jiang and Conrath	0.57	3	1	0.29	2	10	19	0.71
- Lin	0.51	2	1	0.37	0	10	16	0.83
Distributional items	0.64	6	1	0.29	1	6	27	0.79
Phonetic items	0.60	4	0	0.34	0	5	31	0.66
Mixed strategy items	0.53	1	0	0.39	0	5	14	0.89

Table 2: Item analysis

lower and upper groups to be 0.74 for the sets of distractors generated using collocations. For the WordNet distractors the results were as follows. The average distance between the students in the lower and upper groups was found to be 0.71 for the extended gloss overlap distractors, 0.76 for the Leacock and Chodorow distractors, 0.71 for the Jiang and Conrath distractors and 0.83 for the Lin distractors. For the distractors selected by way of distributional similarity the average difference between students in the lower and upper groups was 0.79, for the phonetic distractors — 0.66 and for those selected by a mixed strategy — 0.89.

In our evaluation we also used the notions of *poor distractors* as well as *not-useful* distractors. Distractors are classed as *poor* if they attract more students from the upper group than from the lower group. There were 2 (2.5%) poor distractors from the collocation distractors. The WordNet distractors fared as follows with regard to the number of poor distractors. There were altogether 9 (11%) poor distractors from the extended gloss overlap distractors, 9 (11%) from the Leacock and Chodorow distractors, 10 (12%) from the Jiang and Conrath distractors and 10 (12%) from the Lin ones. There were 6 (7.5%) from the distributional similarity which were classed as poor, 5 (6%) from the phonetic similarity ones were classed as poor and 5 (6%) from the distractors selected through a mixed strategy were classed as such (Table 2).

On the other hand, distractors are termed *not useful* if they are not selected by any students at all. The evaluation showed (see Table 2) that there were 24 (30%) distractors deemed not useful from the collocation distractors. The figures for not useful distractors for those selected by way of WordNet similarity were as follows: 17 (21%) for extended gloss overlap distractors, 20 (25%) for the Leacock and Chodorow distractors, 19 (24%) for the Jiang and Conrath distractors and 16 (20%) for the Lin ones. From the distributional distractors, 27 (34%) emerged as not useful, whereas 31 (39%) phonetic similarity and 14 (18%) mixed strategy distractors were found not useful.

The overall figures suggest that the 'most useful' distractors are those chosen with mixed strategy (highest average difference 0.89; lowest number of not useful distractors, second lowest number of poor distractors), followed by those chosen with Lin's WordNet measure (second highest average distance of 0.83; second lowest number of not useful distractors).

Summarising the results of the item analysis, it is clear that there is not a method that outperforms the rest in terms of producing best quality items or distractors. At the same time it is also clear that in general the mixed strategy and Lin's measure consistently perform better than the rest of methods/measures. Phonetic similarity did not deliver as expected.

Although the results indicate that the Lin items have the best average item difficulty, none of the difference (between item difficulty of Lin and other methods, or between any pair of methods) is statistically significant. From the DP point of view, only the difference between mixed strategy (0.39) and distributional items (0.29) is statistically significant ($p<0.05$). For the distractor usefulness measure, none of the difference is statistically significant ($p<0.05$).

5 Conclusion

In this study we conducted extrinsic evaluation of several similarity methods (collocation patterns; four different methods of WordNet-based semantic similarity: extended gloss overlap measure, Leacock and Chodorow's, Jiang and Conrath's as well as Lin's measures; distributional similarity; phonetic similarity; mixed strategy) by seeking to establish which one would be most suitable for the task of selection of distractors in multiple-choice tests. The evaluation results based on item analysis suggests that whereas there is not a method that clearly outperforms in terms of delivering better quality distractors, mixed strategy and Lin's measure consistently perform better than the rest of methods/measures. However, these two methods do not offer any statistically significant improvement over their closest competitors.

Acknowledgments

We would like to express our gratitude to Kathelijne Denturck, Johann Haller, Veronique Hoste, Constantin Orasan, Miriam Seghiri, Andrea Stockero and Irina Temnikova for helping us in the organisation of the in-class experiments.

References

Banerjee, S. and Perderson, T. 2003. Extended gloss overlaps as a measure of semantic relatedness. *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, 805-810.

Dagan I., Lee L., and Pereira F. 1997. Similarity-based methods for word sense disambiguation. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*. Madrid, Spain, 56-63.

Gronlund, N. 1982. *Constructing achievement tests*. New York: Prentice-Hall Inc.

Jiang, J. and Conrath, D. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. *Proceedings of the International Conference on Research in Computational Linguistics. Taiwan*, 19-33.

Lin, D. 1997. Using syntactic dependency as a local context to resolve word sense ambiguity. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*. Madrid, Spain, 4-71.

Leacock, C., Chodorow, M. 1998. Combining local context and WordNet similarity for word sense identification. In: Fellbaum, C., 1998, *WordNet: An Electronic Lexical Database.* MIT Press, Cambridge, MA, 265–283.

Mitkov R. and Ha L.A. 2003. Computer-aided generation of multiple-choice tests. *Proceedings of the HLT/NAACL 2003 Workshop on Building educational applications using Natural Language Processing*. Edmonton, Canada, 17-22.

Mitkov, R., An, L.A. and Karamanis, N. 2006. "A computer-aided environment for generating multiple-choice test items". *Journal of Natural Language Engineering,* 12 (2): 177-194.

Patwardhan, S, Banerjee, S. and Pedersen, T. 2003. Using Measures of Semantic Relatedness for Word Sense Disambiguation. *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics.* Mexico City, 241-257.

Resnik, P. 1995. Using information content to evaluate semantic similarity in a taxonomy. *Proceedings of the 14th International Joint Conference on Artificial Intelligence.* Montreal, 448–453.

Tapanainen, P. and Järvinen, T. 1997. A non-projective dependency parser. *Proceedings of the 5th Conference of Applied Natural Language Processing,* Washington, 64-71.

Vajda, E.J. 2001 Course Materials from the module of Introduction to Linguistics. Professor Edward J. Vajda Homepage, Washington, Western Washington University, Modern and Classical Languages.http://pandora.cii.wwu.edu/vajda/ling201/ling201home.htm

Paraphrase assessment in structured vector space: Exploring parameters and datasets

Katrin Erk
Department of Linguistics
University of Texas at Austin
katrin.erk@mail.utexas.edu

Sebastian Padó
Department of Linguistics
Stanford University
pado@stanford.edu

Abstract

The appropriateness of paraphrases for words depends often on context: "grab" can replace "catch" in "catch a ball", but not in "catch a cold". Structured Vector Space (SVS) (Erk and Padó, 2008) is a model that computes *word meaning in context* in order to assess the appropriateness of such paraphrases. This paper investigates "best-practice" parameter settings for SVS, and it presents a method to obtain large datasets for paraphrase assessment from corpora with WSD annotation.

1 Introduction

The meaning of individual occurrences or *tokens* of a word can change vastly according to its context. A central challenge for computational lexical semantics is describe these *token meanings* and how they can be computed for new occurrences.

One prominent approach to this question is the *dictionary-based model* of token meaning: The different meanings of a word are a set of distinct, disjoint senses enumerated in a lexicon or ontology, such as WordNet. For each new occurrence, determining token meaning means choosing one of the senses, a classification task known as Word Sense Disambiguation (WSD). Unfortunately, this task has turned out to be very hard both for human annotators and for machines (Kilgarriff and Rosenzweig, 2000), not at least due to granularity problems with available resources (Palmer et al., 2007; McCarthy, 2006). Some researchers have gone so far as to suggest fundamental problems with the concept of categorical word senses (Kilgarriff, 1997; Hanks, 2000).

An interesting alternative is offered by *vector space models* of word meaning (Lund and Burgess, 1996; McDonald and Brew, 2004) which characterize the meaning of a word entirely without reference to word senses. Word meaning is described in terms of a vector in a high-dimensional vector space that is constructed with distributional methods. Semantic similarity is then simply distance to vectors of other words. Vector space models have been most successful in modeling the meaning of word types (i.e. in constructing *type vectors*). The characterization of token meaning by corresponding *token vectors* would represent a very interesting alternative to dictionary-based methods by providing a direct, graded, unsupervised measure of (dis-)similarity between words in context that completely avoids reference to dictionary senses. However, there are still considerable theoretical and practical problems, even though there is a substantial body of work (Landauer and Dumais, 1997; Schütze, 1998; Kintsch, 2001; Mitchell and Lapata, 2008).

In a recent paper (Erk and Padó, 2008), we have introduced the *structured vector space (SVS)* model which addresses this challenge. It yields one token vector per input word. Token vectors are not computed by combining the lexical meaning of the surrounding words – which risks resulting in a "topicality" vector – but by modifying the type meaning of a word with the semantic expectations of syntactically related words, which can be thought of as selectional preferences. For example, in *catch a ball*, the token vector for *ball* is computed by combining the type vector of *ball* with a vector for the *selectional preferences* of *catch* for its object. The token vector for *catch*, conversely, is constructed from the type vector of *catch* and the *inverse object preference vector* of *ball*. The resulting token vectors describe the meaning of a word in a particular sentence not through a sense label, but through the distance of the token vector to other vectors.

A natural question that arises is how vector-based models of token meaning can be evaluated. It is of course possible to apply them to a traditional WSD task. However, this strategy remains vulnerable to all criticism concerning the annotation of categorical word senses, and also does not take advantage of the vector models' central asset, namely gradedness. Thus, *paraphrase-based assessment for models of token meaning* was proposed as a representation-neutral disambiguation task that can replace WSD (McCarthy and Navigli, 2007; Mitchell and Lapata, 2008). Given a word token in context and a set of potential paraphrases, the task consists of identifying the *subset* of valid paraphrases. For example, in the following example, the first paraphrase is appropriate, but the second is not:

(1) Google *acquired* YouTube ⇒
Google *bought* YouTube

(2) How children *acquire* skills ⇏
How children *buy* skills

This task is graded in the sense that there is no disjoint set of labels from which exactly one is picked for each token; rather, the paraphrases form a set of labels of which a subset is appropriate for each word token,

Proceedings of the EACL 2009 Workshop on GEMS: GEometical Models of Natural Language Semantics, pages 57–65,
Athens, Greece, 31 March 2009.

and the appropriate sets for two tokens may overlap to varying degrees. In an ideal vector-based model, valid paraphrases such as (1) should possess similar vectors, and invalid ones such as (2) dissimilar ones.

In Erk and Padó (2008), we evaluated SVS on two variants of the paraphrase assessment test: first, the prediction of human judgments on a seven-point scale for paraphrases for verb-subject pairs (Mitchell and Lapata, 2008); and second, the original Lexical Substitution task by McCarthy and Navigli (2007). To avoid overfitting, we optimized our parameters on the first dataset and evaluated only the best model on the second dataset. However, given evidence for substantial inter-task differences, it is unclear to what extent these parameters are optimal beyond the Mitchell and Lapata dataset. This paper addresses this question with two experiments:

Impact of parameters. We re-examine three central parameters of SVS. The first one is the choice of *vector combination function*. Following Mitchell and Lapata (2008), we previously used componentwise multiplication, whose interpretation in vector space is not straightforward. The second one is *reweighting*. We obtained the best performance when the context expectations were reweighted by taking each component to a (high) n-th power, which is counterintuitive. Finally, we found subjects to be more informative in judging the appropriateness of paraphrases than objects. This appears to contradict work in theoretical syntax (Levin and Rappaport Hovav, 2005).

To reassess the role of these parameters, we construct a controlled dataset of transitive instances from the Lexical Substitution corpus to reexamine and investigate these issues, with the aim of providing "best practice" settings for SVS. This turns out to be more difficult than expected, leading us to suspect that a globally optimal parameter setting across tasks may simply not exist. We also test a simple extension of SVS that uses a richer context (both subject and object) to construct the token vector, with first positive results.

Dataset creation. The Lexical Substitution dataset used in Erk and Padó (2008) was very small, which limits the conclusions that can be drawn from it. This points towards a more general problem of paraphrase-based assessment for models of token meaning: Until now, all datasets for this task were specifically created by hand. It would provide a strong boost for paraphrase assessment if the large annotated corpora that are available for WSD could be reused.

We present an experiment on converting the WordNet-annotated SemCor corpus into a set of "pseudo-paraphrases" for paraphrase-based assessment. We use the synonyms and direct hypernyms of an annotated synset as these "pseudo-paraphrases". While the synonyms and hypernyms are not guaranteed to work as direct replacements of the target word in the given context, they are semantically similar to the target word. The result is a dataset ten times larger than the Lex-Sub dataset. As we describe in this paper, we find that this method is nevertheless problematic: The resulting dataset is considerably more difficult to model than the existing hand-built paraphrase corpora, and its properties differ considerably from the manually constructed Lexical Substitution dataset.

2 The structured vector space model

The main intuition behind the SVS model is to treat the interpretation of a word in context as guided by *expectations about typical events*. This move to include typical arguments and predicates into a model of word meaning is motivated both on cognitive and linguistic grounds. In cognitive science, the central role of expectations about typical events on almost all aspects of human language processing is well-established (McRae et al., 1998; Narayanan and Jurafsky, 2002). In linguistics, expectations have long been used in semantic theories in the form of *selectional restrictions* and *selectional preferences* (Wilks, 1975), and more recently induced from corpora (Resnik, 1996). Attention has mostly been limited to selectional preferences of verbs, which have been used for for a variety of tasks (Hindle and Rooth, 1993; Gildea and Jurafsky, 2002). A recent result that the SVS model builds on is that selectional preferences can be represented as *prototype* vectors constructed from seen arguments (Erk, 2007; Padó et al., 2007).

Representing lemma meaning. To accommodate information about semantic expectations, the SVS model extends the traditional representation of word meaning as a single vector by a set of vectors, each of which represents the word's *selectional preferences* for each relation that the word can assume in its linguistic context. While we ultimately think of these relations as "properly semantic" in the sense of semantic roles, the instantiation of SVS we consider in this paper makes use of dependency relations as a level of representation that generalizes over a substantial amount of surface variation but that can be obtained automatically with high accuracy using current NLP tools.

The idea is illustrated in Figure 1. In the representation of the verb *catch*, the central square stands for the lexical vector of *catch* itself. The three arrows link it to *catch*'s preferences for dependency relations it can participate in, such as for its $subj$ects, its objects, and for verbs for which it appears as a complement ($comp^{-1}$). The figure shows the head words that enter into the computation of the selectional preference vector. Likewise, *ball* is represented by one vector for *ball* itself, one for *ball*'s preferences for its modifiers (mod), and two for the verbs of which it can occur as a subject ($subj^{-1}$) and an object (obj^{-1}), respectively.

This representation includes selectional preferences (like $subj$, obj, mod) exactly parallel to *inverse* selectional preferences ($subj^{-1}$, obj^{-1}, $comp^{-1}$). The SVS model is then formalized as follows. Let D be a vector space, and let $\mathcal{R}$ be some set of relation labels. We then

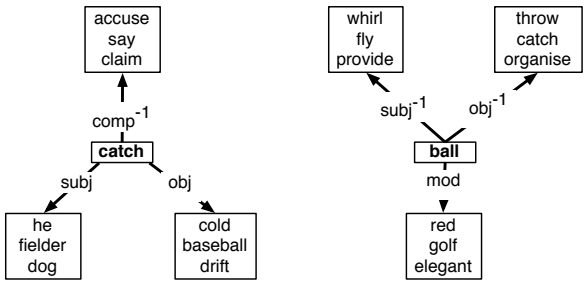


Figure 1: Structured Vector Space representations for noun *ball* and verb *catch*: Each box represents one vector (lexical information or expectations)

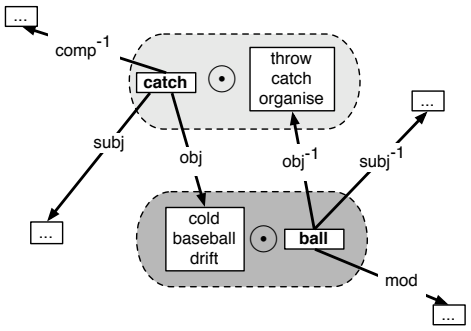


Figure 2: Combining predicate and argument via relation-specific semantic expectations

represent the meaning of a lemma w as a triple

$$m(w) = (v_w, R, R^{-1})$$

where $v_w \in D$ is the type vector of the word w itself, $R : \mathcal{R} \rightarrow D$ maps each relation label onto a vector that describes w's selectional preferences, and $R^{-1} : \mathcal{R} \rightarrow D$ maps from role labels to vectors describing inverse selectional preferences of w. Both R and R^{-1} are partial functions. For example, the direct object preference is undefined for intransitive verbs.[1]

Computing meaning in context. SVS computes the meaning of a word a in the context of another word b via their selectional preferences as follows: Let $m(a) = (v_a, R_a, R_a^{-1})$ and $m(b) = (v_b, R_b, R_b^{-1})$ be the representations of the two words, and let $r \in \mathcal{R}$ be the relation linking a to b. Then, the meaning of a and b in this context is defined as a pair of structured vector triples: $m(a \xrightarrow{r} b)$ is the meaning of a with b as its r-argument, and $m(b \xrightarrow{r^{-1}} a)$ the meaning of b as the r-argument of a:

$$\begin{aligned} m(a \xrightarrow{r} b) &= \left(v_a \odot R_b^{-1}(r), R_a - \{r\}, R_a^{-1}\right) \\ m(b \xrightarrow{r^{-1}} a) &= \left(v_b \odot R_a(r), R_b, R_b^{-1} - \{r\}\right) \end{aligned} \tag{3}$$

where $v_1 \odot v_2$ is a direct vector combination function as in traditional models, e.g. addition or component-wise multiplication. If either $R_a(r)$ or $R_b^{-1}(r)$ are not defined, the combination fails. Afterward, the filled argument position r is deleted from R_a and R_b^{-1}.

Figure 2 illustrates the procedure on the representations from Figure 1. The dotted lines indicate that the lexical vector for *catch* is combined with the inverse object preference of *ball*. Likewise, the lexical vector for *ball* combines with the object preference vector of *catch*.

Recursive application. In Erk and Padó (2008), we considered only one combination step; however, the syntactic context of a word in a dependency tree often consists of more than one word. It seems intuitively plausible that disambiguation should profit from more context information. Thus, we extend SVS with recursive application. Let a stand in relation r to b. As defined above, the result of combining $m(a)$ and $m(b)$ by relation r are two structured vector triples $m(a \xrightarrow{r} b)$ and $m(b \xrightarrow{r^{-1}} a)$. If a also stands in relation $s \neq r$ to a word c with $m(c) = (v_a, R_a, R_a^{-1})$, we define the meaning of a in the context of b and c canonically as

$$\begin{aligned} m(m(a \xrightarrow{r} b) \xrightarrow{s} c) &= \big((v_a \odot R_b^{-1}(r)) \odot R_c^{-1}(s), \\ &\quad R_a - \{r, s\}, R_a^{-1}\big) \end{aligned} \tag{4}$$

If $\odot$ is associative and commutative, then $m(m(a \xrightarrow{r} b) \xrightarrow{s} c) = m(m(a \xrightarrow{s} c) \xrightarrow{r} b)$. This will be the case for all the combination functions we use in this paper.

Note that this is a simplistic model of the influence of multiple context words: it computes only lexical meaning recursively, but does not model the influence of context on the *selectional preferences*. For example, the subject selectional preferences of *catch* are identical to those of *catch the ball*, even though one would expect that the *outfielder* corresponds much better to the expectations of *catch the ball* than of just *catch*.

3 Experimental Setup

The task that we are considering is *paraphrase assessment in context*. Given a predicate-argument pair and a paraphrase candidate, the models have to decide how appropriate the paraphrase is for the predicate-argument combination. This is the main task against which token vector models have been evaluated in the past (Mitchell and Lapata, 2008; Erk and Padó, 2008). In Experiment 1, we use manually created paraphrases. In Experiment 2, we replaces human-generated paraphrases with "pseudo-paraphrases", contextually similar words that may not be completely appropriate as paraphrases in the given context, but can be collected automatically. Our parameter choices for SVS are as similar as possible to the second experiment of our earlier paper.

Vector space. We use a dependency-based vector space that counts a target word and a context word

[1] We use separate functions R, R^{-1} rather than a joint syntactic context preference function because (a) this separation models the conceptual difference between predicates and arguments, and (b) it allows for a simpler, more elegant formulation of the computation of meaning in context in Eq. 3.

as co-occurring in a sentence if they are connected by an "informative" path in the dependency graph for the sentence.[2] We build the space from a Minipar-parsed version of the British National Corpus with dependency parses obtained from Minipar (Lin, 1993). It uses raw co-occurrence counts and 2000 dimensions.

Selectional preferences and reweighting. We use a prototype-based selectional preference model (Erk, 2007). It models the selectional preferences of a predicate for an argument position as the weighted centroid of the vectors for all head words seen for this position in a large corpus. Let $f(a, r, b)$ denote the frequency of a occurring in relation r to b in the parsed BNC. Then, we compute the selectional preferences as:

$$R'_b(r) = \frac{1}{N} \sum_{a: f(a,r,b)>0} f(a, r, b) \cdot \vec{v}_a \qquad (5)$$

where N is the number of fillers a with $f(a, r, b) > 0$.

In Erk and Padó (2008), we found that applying a *reweighting* step to the selectional preference vector by taking each component of the centroid vector $R'_b(r)$ to the n-th power lead to substantial improvements. The motivation for this technique is to alleviate noise arising from the use of unfiltered head words for the construction. The reweighted selectional preference vector $R_b(r)$ is defined as:

$$R_b(r) = \langle v_1^n, \ldots, v_m^n \rangle \text{ for } R'_b(r) = \langle v_1, \ldots, v_m \rangle \quad (6)$$

where we write $\langle v_1, \ldots, v_m \rangle$ for the sequence of values that make up a vector $R'_b(r)$. Inverse selectional preferences $R_b^{-1}(r)$ of nouns are defined analogously, by computing the centroid of the verbs seen as governors of the noun in relation r.

In this paper, we test reweighting parameters of n between 0.5 and 30. Generally, small ns will decrease the influence of the selectional preference vector. The result can be thought of as a "word type vector modified by context expectations", while large ns increase the role of context, until we arrive at a "contextual expectation vector modified by the word type vector". [3]

Vector combination. We test three vector combination functions $\odot$, which have different interpretations in vector space. The simplest one is componentwise addition, abbreviated as **add**, i.e., simple vector addition.[4] With addition, context dimensions receive a high count whenever either of the two vectors has a high co-occurrence count for the context.

[2] We used the minimal context specification and plain weight of the DependencyVectors software package.

[3] For the component-wise minimum combination (see below), where we normalize the vectors before the combination, the reweighting has a different effect. It shifts most of the mass onto the largest-value dimensions and sets smaller dimensions to values close to zero.

[4] Since we subsequently focus on cosine similarity, which is length-invariant, vector addition can also be interpreted as centroid computation.

Next, we test component-wise multiplication (**mult**). This operation is more difficult to interpret in terms of vector space, since it does not correspond to the standard inner or outer vector products. The most straightforward interpretation is to reinterpret the second vector as a diagonal matrix, i.e., as a linear transformation of the first vector. Large entries in the second vector increase the weight of the corresponding contexts; small entries decrease it. Mitchell and Lapata (2008) found this method to yield the best results.

The third vector combination function we consider is component-wise minimum (**min**). This combination function results in a vector with high counts only for contexts which co-occur frequently with both input vectors and can thus be understood as an intersection between the two context sets. Since the entries of two vectors need to be on the same order to magnitude for this method to yield meaningful results, we normalize vectors before the combination for **min**.

Assessing models of token meaning. Given a transitive verb v with subject a and direct object b, we test three variants of computing a token vector for v. The first two involve only one combination step. In the **subj** condition, v's type vector is combined with the inverse subject preference vector of a. In the **obj** condition, v's type vector is combined with the inverse object preference vector of b. The third variant is the recursive application of the SVS combination procedure described in Section 2 (condition **both**). Specifically, we combine v's type vector with both a's inverse subject preference and with b's inverse object preference to obtain a "richer" token vector.

In all three cases, the resulting token vector is compared to the *type* vector of the paraphrase (in Experiment 1) or the semantically related word (in Experiment 2). We use Cosine Similarity, a standard choice as vector space similarity measure.

4 Experiments

4.1 Experiment 1: The impact of parameters

In our 2008 paper, we tested the LexSub data only with the parameters that showed best results on the Mitchell and Lapata data: vector combination using componentwise multiplication (*mult*), and the computation of (inverse) selectional preference vectors with high powers of $n = 20$ or $n = 30$. However, there were indications that the two datasets showed fundamental differences. In particular, the Mitchell and Lapata data could only be modeled using a PMI-transformed vector space, while the LexSub data could only be modeled using raw co-occurrence count vectors.

Another one of our findings that warrants further inquiry stems from our comparison of different context choices (verb plus subject, verb plus object, noun plus embedding verb). We found that subjects are better disambiguators than objects. This seems counterintuitive both on theoretical and empirical grounds. Theoretically,

Sentence	Substitutes
By asking people who **work** there, I have since determined that he didn't. (# 2002)	be employed 4; labour 1
Remember how hard your ancestors **worked**. (# 2005)	toil 4; labour 3; task 1

Figure 3: Lexical substitution example items for "work"

		0.5	1	2	5	10	20
add	obj	**61.5**	59.7	58.9	56.1	56.0	55.7
add	subj	**61.7**	**61.7**	59.5	58.4	57.3	57.0
add	both	**61.3**	60.0	60.2	57.7	57.1	56.7
mult	obj	**59.8**	59.7	57.8	55.7	55.7	55.4
mult	subj	**60.3**	59.7	59.3	57.3	57.7	56.7
mult	both	**59.9**	58.8	57.1	55.8	55.3	$<1^{\text{Pr}}$
min	obj	**60.2**	60.0	59.5	57.3	55.7	55.8
min	subj	**62.2**	60.5	59.1	58.5	57.8	57.0
min	both	**62.3**	60.2	59.8	57.3	55.8	55.1

Table 1: OOT accuracy on the LEXSUB-PARA dataset across models and reweighting values (best results for each model boldfaced). Random baseline: 53.7. Target type vector baseline: 57.1. $^{\text{Pr}}$: Numerical problem.

the notion of verb phrase has been motivated, among other things, with the claim that direct objects contribute more to a verb's disambiguation than subjects (Levin and Rappaport Hovav, 2005). Empirically, subjects are known to be realized more often as pronouns than objects, which makes their vector representations less semantically specific. However, we used *two different datasets* – the subject results on a set of intransitive verbs, and the object results on a set of transitive verbs, so the results are not comparable.

In this experiment, we construct a new, more controlled dataset from the Lexical Substitution corpus to systematically assess the importance of the three main parameters: the relation used for disambiguation, the combination function, and the reweighting parameter.

Construction of the LEXSUB-PARA dataset. The original Lexical Substitution corpus, constructed for the SemEval-1 lexical substitution task (McCarthy and Navigli, 2007), consists of 10 instances each of 200 target words in sentential contexts, drawn from a large internet corpus (Sharoff, 2006). Contextually appropriate paraphrases for each instance of each target word were elicited from up to 6 participants. Figure 3 shows two instances for the verb *to work*. The frequency distribution over paraphrases can be understood as a characterization of the target word's meaning in each context.

For the current paper, we constructed a new subset of LexSub we call LEXSUB-PARA by parsing LexSub with Minipar (Lin, 1993) and extracting all 177 sentences with transitive verbs that had overtly realized subjects and objects, regardless of voice. We did not manually verify the correctness of the parses, but discarded 17 sentences where we were not able to compute inverse selectional preferences for the subject or object head word (these were mostly rare proper names). This left 160 transitive instances of 42 verbs.

Evaluation For evaluation, we use a variant of the SemEval "out of ten" (OOT) evaluation metrics defined by McCarthy and Navigli (2007). They developed two metrics, OOT Precision and Recall, which compare where a predicted set of appropriate paraphrases must be evaluated against a gold standard set. Their metrics are called "out of ten" because they are measure the accuracy of the first ten paraphrases predicted by the system. Since they allow systems to abstain from predictions for any number of tokens, their two variants average this accuracy (a), over the tokens with a prediction (OOT Precision), and (b), over all tokens (OOT Recall). Since our system produces predictions for all tokens, OOT Precision and Recall become identical.

Formally, let G_i be the gold paraphrases for occurrence i, and let $f(s,i)$ be the frequency with which s has been named as paraphrase for i. Let M_i be the ten paraphrase candidates top-ranked by the SVS model for i. We write out-of-ten accuracy (OOT) as:

$$\text{OOT} = 1/|I| \sum_i \frac{\sum_{s \in M_i \cap G_i} f(s,i)}{\sum_{s \in G_i} f(s,i)} \qquad (7)$$

We compute two baselines. The first one is random baseline that guesses whether paraphrases are appropriate. The second baseline uses the original type vector of the target verb without any combination, i.e., its "out of context meaning", as representation for the token.

Results. Table 1 shows the results on the LEXSUB-PARA dataset. Recall that the task is to decide the appropriateness of paraphrases for verb instances, disambiguated by the inverse selectional preferences of their subjects (*subj*), their objects (*obj*), and *both*. The random baseline attains an OOT accuracy of 53.7, and the type vector of the target vector performs at 57.1.

SVS is able to outperform both baselines for all values of the reweighting parameter n <2, and we find the best results for the lowest value, $n = 0.5$. As for the influence of the vector combination function, the best result is yielded by **min** (OOT=62.3), followed by **add** (OOT=61.7), while **mult** shows generally worse results (OOT=60.3). For both **add** and **mult**, using only the subject as context only is optimal. The overall best result, using **min**, is seen for *both*; however, the improvement over *subj* is very small.

In the model **mult**-*both*-20, where target vectors were multiplied with two very large expectation vectors, almost all instances failed due to overflow errors.

Discussion. Our results indicate that our parameter optimization strategy in Erk and Padó (2008) was in fact flawed. The parameters that were best for the Mitchell and Lapata (2008) data (**mult**, $n = 20$) are suboptimal for LEXSUB-PARA data.[5] The good results for low val-

[5]We assume that our results hold for the Padó & Erk (2008) lexical substitution dataset as well, due to its similar nature.

ues of n indicate that good discrimination between valid and invalid paraphrases can be obtained by relatively small modifications of the target vector in the direction indicated by the context. Surprisingly, we still find that the results in the *subj* condition are almost always better than those in the *obj* condition, even though the dataset consists only of transitive verbs, where we would have expected the inverse result. We have two partial explanations. First, we find that pronouns, which occur frequently in subject position (*I*, *he*), are still informative enough to distinguish "animate" from "inanimate" paraphrases of verbs such as *touch*. Second, we see a higher number of Minipar errors in for object positions than for subject positions, and consequently more data both for object fillers and for object selectional preferences.

The overall best result was yielded by a condition that used *both* (subject plus object) for disambiguation, using the recursive modification from Eq. (4). While we see this as a promising result, the difference to the second-best result is very small, in almost all other conditions the performance of *both* is close to the average of *obj* and *subj* and thus a suboptimal choice.

4.2 Experiment 2: Creating larger datasets with pseudo-paraphrases

With a size of 2,000 sentences, even the complete LexSub dataset is tiny in comparison to many other resources in NLP. Limiting attention to successfully parsed transitive instances results in an even smaller dataset on which it is difficult to distinguish noise from genuine differences between models. This is a large problem for the use of paraphrase appropriateness as evaluation task for models of word meaning in context.

In consequence, the automatic creation of larger datasets is an important task. While unsupervised methods for paraphrase induction are becoming available (e.g., Callison-Burch (2008)), they are still so noisy that the created datasets cannot serve as gold standards. However, there is an alternative strategy: there is a considerable amount of data in different languages annotated with *categorical* word sense, created (e.g.) for Word Sense Disambiguation exercises such as Senseval. We suggest to convert these data for use in a task similar to paraphrase assessment, interpreting available information about the word sense as *pseudo-paraphrases*. Of course, the caveat is that these pseudo-paraphrases may behave differently than genuine paraphrases. To investigate this issue, we repeat Experiment 1 on this dataset.

Construction of the SEMCOR-PARA dataset The SemCor corpus is a subset of the Brown corpus that contains 23,346 lemmas annotated with senses according to WordNet 1.6. Fortunately, WordNet provides a rich characterization of word senses. This allows us to use the WordNet *synonyms* of a given word sense as pseudo-paraphrases. Since it can be the case that the target word is the only word in a synset, we also need to add *direct hypernyms*. Direct hypernyms have been used in annotation tasks to characterize WordNet senses (Mihalcea and Chklovski, 2003), an indicator that they are usually close enough in meaning to function as pseudo-paraphrases.

		0.5	1	2	5	10	20
add	obj	21.7	20.7	23.2	**24.3**	24.2	21.8
add	subj	20.6	20.1	22.9	**24.4**	23.3	19.7
add	both	21.1	20.3	23.2	**24.4**	23.3	18.9
mult	obj	22.6	24.8	**25.0**	24.4	24.2	21.4
mult	subj	21.1	23.9	**24.4**	**24.4**	23.5	19.8
mult	both	24.5	24.5	**25.6**	24.3	20.0	17.4
min	obj	20.9	19.5	23.6	**24.4**	24.3	21.9
min	subj	20.1	19.6	22.5	**24.2**	23.9	19.6
min	both	20.1	19.8	**25.2**	24.5	24.3	19.0

Table 2: OOT accuracy on the SEMCOR-PARA dataset across models and reweighting values (best results for each line boldfaced). Random baseline: 19.6. Target type vector baseline: 20.8

Again, we parsed the corpus with Minipar and identified all sense-tagged instances of the verbs from LEXSUB-PARA, to keep the two corpora as comparable as possible. For each instance w_i of word w, we collected all synonyms and direct hypernyms of the synset as the set of appropriate paraphrases. The list of synonyms and direct hypernyms of all other senses of w, whether they occur in SemCor or not, were considered inappropriate paraphrases for the instance w_i. This method does not provide us with frequencies for the pseudo-paraphrases; we thus assumed a uniform frequency of 1. This does not do away with the gradedness of the meaning representation, though, since each token is still associated with a set of appropriate paraphrases.

Out of 2242 transitive verb instances, we further removed 153 since we could not compute selectional preferences for at least one of the fillers. 484 instances were removed because WordNet did not list any verbal paraphrases for the annotated synset or its direct hypernym. This resulted in 1605 instances for 40 verbs, a dataset an order of magnitude larger than LEXSUB-PARA. (See Section 4.3 for an example verb with paraphrases.)

Results and Discussion. We again use the OOT accuracy measure. The results for paraphrase assessment on SEMCOR-PARA are shown in Table 2. The numbers are substantially lower than for LEXSUB-PARA. This is first and foremost a consequence of the higher "polysemy" of the pseudo-paraphrases. In LEXSUB-PARA, the average numbers of possible paraphrases per target word is 20; in SEMCOR-PARA, 54. This is to be expected and also reflected in the much lower random baseline (19.6% OOT). However, we also observe that the reduction in error rate over the baseline is considerably lower for SEMCOR-PARA than for LEXSUB-PARA (10% vs. 20% reduction).

Among the parameters of the model, we find the largest impact for the reweighting parameter. The best results occur in the middle range($n = 2$ and $n = 5$),

with both lower and higher weights yielding considerably lower scores. Apparently, it is more difficult to strike the right balance between the target and the expectations on this dataset. This is also mirrored in the smaller improvement of the target type vector baseline over the random baseline. As for vector combination functions, we find the best results for the more "intersection"-like **mult** and **min** combinations, with somewhat lower results for **add**; however, the differences are rather small. Finally, combination with *obj* works better than combination with *subj*. At least among the best results, *both* is able to improve over the use of either individual relation. The best result uses **mult**-*both*, with an OOT accuracy of 25.6.

4.3 Further analysis

In our two experiments, we have found systematic relationships between the SVS model parameters and their performance within the LEXSUB-PARA and SEMCOR-PARA datasets. Unfortunately, few of the parameter settings we found to work well appear to generalize across the two datasets; neither do they correspond to the optimal parameter values we established for the Mitchell and Lapata dataset in our 2008 paper. Variables that vary particularly strikingly are the reweighting parameter and the performance of different relations. To better understand these differences, we perform a further validation analysis that attempts to link model performance to a variable that (a) behaves consistently across the two datasets used in this paper and (b) sheds light onto the patterns we have observed for the parameters.

The quantity we will use for this purpose is the average *discriminativity* of the model. We define discriminativity as the degree to which the token vector computed by the model is on average more similar to the valid than to the invalid paraphrases. For a paraphrase ordering task such as the one we are considering, we want this quantity to be as large as possible; very small quantities indicate that the model is basically "guessing" an order.

Figure 4 plots disciminativity against model performance. As can be expected, it is indeed a very strong correlation between discriminativity and OOT accuracy across all models. A Pearson's correlation test confirms that the correlation is highly significant for both datasets (LEXSUB-PARA: r=0.65, $p < 0.0001$; SEMCOR-PARA: r=0.76, $p < 0.0001$).

Next, we considered the relationship between the mean discriminativity for different combinations and reweighting values n. Figure 5 shows the resulting plots, which reveal two main differences between the datasets. The first one is the influence of the reweighting parameter. For LEXSUB-PARA, the highest discriminativity is found for small values of n, with decreasing values for higher parameter values. In contrast, SEMCOR-PARA shows the highest discriminativity for middle values of n (on the order of 5–10), with lowest values on either side. The second difference is the relative discriminativity of *obj* and *subj*. On LEXSUB-PARA, the *subj* predictions are more discriminative than *obj* predictions for all values of n. On SEMCOR-PARA, this picture is reversed, with more discriminative *obj* predictions for the best (and thus relevant) values of n.

We interpret these patterns, which fit the observed OOT accuracy numbers well, as additional evidence that the variations we see between the datasets are not noise or artifacts of the setup, but arise due to the different makeup of the two datasets. This ties in with our intuitions about the differences between human-generated paraphrases and WordNet "pseudo-paraphrases". Compare the following paraphrase lists:

dismiss (LexSub): banish, deride, discard, discharge, dispatch, excuse, fire, ignore, reject, release, remove, sack

dismiss (SemCor/WordNet): alter, axe, brush, can, change, discount, displace, disregard, dissolve, drop, farewell, fire, force, ignore, modify, notice, packing, push, reject, remove, sack, send, terminate, throw, usher

The SEMCOR-PARA list contains a larger number of unspecific pseudo-paraphrases such as *change, push, send*, which stem from direct WordNet hypernyms of the more specific *dismiss* senses. Presumably, these terms are assigned rather general vectors which the SVS finds difficult to rule out as paraphrases. This lowers the discriminativity of the models, in particular for *subj*, and results in the smaller relative improvement over the baseline we observe for SEMCOR-PARA. This suggests that the usability of word sense-derived datasets in evaluations could be improved by taking depth in the WordNet hierarchy into account when including direct hypernyms among the pseudo-paraphrases.

5 Conclusions

In this paper, we have explored the parameter space for the computation of vector-based representations of token meaning with the SVS model.

Our evaluation scenario was paraphrase assessment. To systematically assess the impact of parameter choice, we created two new controlled datasets. The first one, the LEXSUB-PARA dataset, is a small subset of the Lexical Substitution corpus (McCarthy and Navigli, 2007) that was specifically created for this task. The second dataset, SEMCOR-PARA, which is considerably larger, consists in instances from the SemCor corpus whose WordNet annotation was automatically converted into "pseudo-paraphrase" annotation.[6]

We found a small number of regularities that hold for both datasets: namely, that the reweighting parameter is the most important choice for a SVS model, followed by the relation used as context, while the influence of the vector combination function is comparatively small. Unfortunately, the actual settings of these parameters appeared not to generalize well from one dataset to the other. We have collected evidence that these divergences are not due to noise, but to genuine differences

[6] Both datasets can be obtained from the authors.

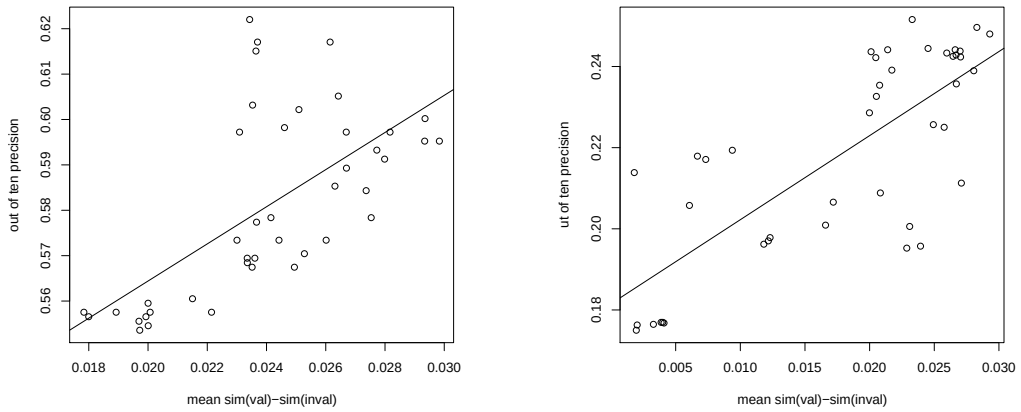


Figure 4: Scatterplot of "out of ten" accuracy against model discriminativity between valid and invalid paraphrases. Left: LEXSUB-PARA, right: SEMCOR-PARA.

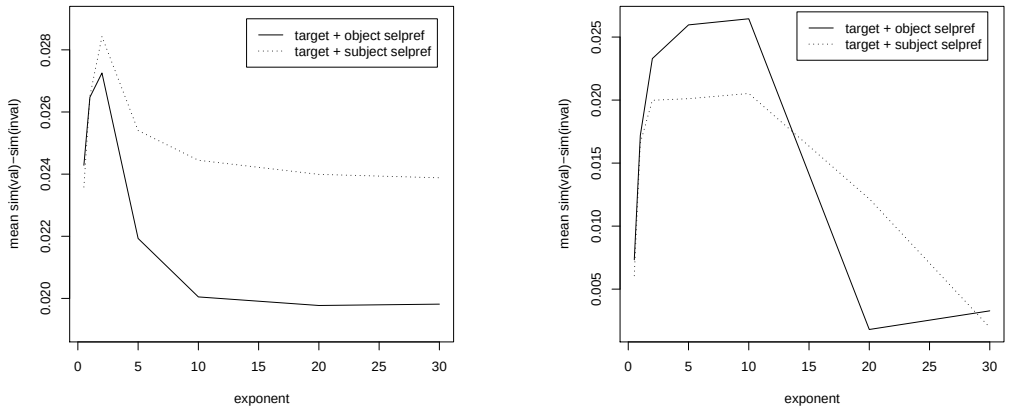


Figure 5: Average amount to which predictions are more similar to valid than to invalid paraphrases, for different reweighting values. Left: LEXSUB-PARA, right: SEMCOR-PARA.

in the datasets. We describe an auxiliary quantity, discriminativity, that measures the ability of the model's predictions to distinguish between valid and invalid paraphrases.

The consequence we draw from this study is that it is surprisingly difficult to establish generalizable "best practice" parameter setting for SVS. Good parameter values appear to be sensitive to the properties of datasets. For example, we have attributed the observation that subjects are more informative on LEXSUB-PARA, while objects work better on SEMCOR-PARA, to differences in the set of paraphrase competitors. In this regard, the conversion of the WSD corpus can be considered a partial success. We have constructed the largest existing paraphrase assessment corpus. However, the use of WordNet information to create paraphrases results in a very difficult corpus. We will investigate methods that exclude overly general hypernyms of the target words as paraphrases to alleviate the problems we see currently.

Discriminativity further suggests that paraphrase assessment can be improved by selectional preference representations that are trained to maximize the distance between valid and invalid paraphrases. Such a representation could be provided by discriminative formulations (Bergsma et al., 2008), or by exemplar-based models that are able to deal better with the ambiguity present in the preferences of very general words.

Another important topic for further research is the computation of token vectors that incorporate more than one context word. The current results we obtain for "*both*" are promising but limited; it appears that the successful integration of multiple context words requires strategies that go beyond simplistic addition or intersection of observed contexts.

References

S. Bergsma, D. Lin, and R. Goebel. 2008. Discriminative learning of selectional preference from unlabeled text. In *Proceedings of EMNLP*, pages 59–68.

C. Callison-Burch. 2008. Syntactic constraints on paraphrases extracted from parallel corpora. In *Proceedings of EMNLP*, pages 196–205.

K. Erk and S. Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of EMNLP*.

K. Erk. 2007. A simple, similarity-based model for selectional preferences. In *Proceedings of ACL*, pages 216–223.

D. Gildea and D. Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

P. Hanks. 2000. Do word meanings exist? *Computers and the Humanities*, 34(1-2):205–215.

D. Hindle and M. Rooth. 1993. Structural ambiguity and lexical relations. *Computational Linguistics*, 19(1):103–120.

A. Kilgarriff and J. Rosenzweig. 2000. Framework and results for English Senseval. *Computers and the Humanities*, 34(1-2).

A. Kilgarriff. 1997. I don't believe in word senses. *Computers and the Humanities*, 31(2):91–113.

W. Kintsch. 2001. Predication. *Cognitive Science*, 25:173–202.

T. Landauer and S. Dumais. 1997. A solution to Platos problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240.

B. Levin and M. Rappaport Hovav. 2005. *Argument Realization*. Research Surveys in Linguistics Series. CUP.

D. Lin. 1993. Principle-based parsing without overgeneration. In *Proceedings of ACL*, pages 112–120.

K. Lund and C. Burgess. 1996. Producing high-dimensional semantic spaces from lexical cooccurrence. *Behavior Research Methods, Instruments, and Computers*, 28.

D. McCarthy and R. Navigli. 2007. SemEval-2007 Task 10: English Lexical Substitution Task. In *Proceedings of SemEval*, pages 48–53.

D. McCarthy. 2006. Relating WordNet senses for word sense disambiguation. In *Proceedings of the ACL Workshop on Making Sense of Sense*, pages 17–24.

S. McDonald and C. Brew. 2004. A distributional model of semantic context effects in lexical processing. In *Proceedings of ACL*, pages 17–24.

K. McRae, M. Spivey-Knowlton, and M. Tanenhaus. 1998. Modeling the influence of thematic fit (and other constraints) in on-line sentence comprehension. *Journal of Memory and Language*, 38:283–312.

R. Mihalcea and T. Chklovski. 2003. Open Mind Word Expert: Creating large annotated data collections with web users' help. In *Proceedings of the EACL 2003 Workshop on Linguistically Annotated Corpora (LINC 2003)*, Budapest, Hungary.

J. Mitchell and M. Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL*, pages 236–244.

S. Narayanan and D. Jurafsky. 2002. A Bayesian model predicts human parse preference and reading time in sentence processing. In *Proceedings of NIPS*, pages 59–65.

S. Padó, U. Padó, and K. Erk. 2007. Flexible, corpus-based modelling of human plausibility judgements. In *Proceedings of EMNLP/CoNLL*, pages 400–409.

M. Palmer, H. Dang, and C. Fellbaum. 2007. Making fine-grained and coarse-grained sense distinctions, both manually and automatically. *Journal of Natural Language Engineering*. To appear.

P. Resnik. 1996. Selectional constraints: An information-theoretic model and its computational realization. *Cognition*, 61:127–159.

H. Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–124.

Serge Sharoff. 2006. Open-source corpora: Using the net to fish for linguistic data. *International Journal of Corpus Linguistics*, 11(4):435–462.

Y. Wilks. 1975. Preference semantics. In *Formal Semantics of Natural Language*. CUP.

SVD Feature Selection for Probabilistic Taxonomy Learning

Fallucchi Francesca
Disp, University "Tor Vergata"
Rome, Italy
fallucchi@info.uniroma2.it

Fabio Massimo Zanzotto
Disp, University "Tor Vergata"
Rome, Italy
zanzotto@info.uniroma2.it

Abstract

In this paper, we propose a novel way to include unsupervised feature selection methods in probabilistic taxonomy learning models. We leverage on the computation of logistic regression to exploit unsupervised feature selection of singular value decomposition (SVD). Experiments show that this way of using SVD for feature selection positively affects performances.

1 Introduction

Taxonomies are extremely important knowledge repositories in a variety of applications for natural language processing and knowledge representation. Yet, manually built taxonomies such as WordNet (Miller, 1995) often lack in coverage when used in specific knowledge domains. Automatically creating or extending taxonomies for specific domains is then a very interesting area of research (O'Sullivan et al., 1995; Magnini and Speranza, 2001; Snow et al., 2006). Automatic methods for learning taxonomies from corpora often use distributional hypothesis (Harris, 1964) and exploit some induced lexical-syntactic patterns (Hearst, 1992; Pantel and Pennacchiotti, 2006). In these models, within a very large set, candidate word pairs are selected as new word pairs in hyperonymy and added to an existing taxonomy. Candidate pairs are represented in some feature space. Often, these feature spaces are huge and, then, models may take into consideration noisy features.

In machine learning, feature selection has been often used to reduce the dimensions in huge feature spaces. This has many advantages, e.g., reducing the computational cost and improving performances by removing noisy features (Guyon and Elisseeff, 2003).

In this paper, we propose a novel way to include unsupervised feature selection methods in probabilistic taxonomy learning models. Given the probabilistic taxonomy learning model introduced by (Snow et al., 2006), we leverage on the computation of logistic regression to exploit singular value decomposition (SVD) as unsupervised feature selection. SVD is used to compute the pseudo-inverse matrix needed in logistic regression.

To describe our idea, we firstly review how SVD can be used as unsupervised feature selection (Sec. 2). In Section 3 we then describe the probabilistic taxonomy learning model introduced by (Snow et al., 2006). We will then shortly review the logistic regression used to compute the taxonomy learning model to describe where SVD can be naturally used. We will describe our experiments in Sec. 4. Finally, we will draw some conclusions and describe our future work (Sec. 5).

2 Unsupervised feature selection with Singular Value Decomposition

Singular value decomposition (SVD) is one of the possible factorization of a rectangular matrix that has been largely used in information retrieval for reducing the dimension of the document vector space (Deerwester et al., 1990).

The decomposition can be defined as follows. Given a generic rectangular $n \times m$ matrix A, its singular value decomposition is:

$$A = U\Sigma V^T$$

where U is a matrix $n \times r$, V^T is a $r \times m$ and Σ is a diagonal matrix $r \times r$. The two matrices U and V are unitary, i.e., $U^TU = I$ and $V^TV = I$. The diagonal elements of the Σ are the *singular values* such as $\delta_1 \geq \delta_2 \geq ... \geq \delta_r > 0$ where r is the rank of the matrix A. For the decomposition, SVD exploits the linear combination of rows and columns of A.

A first trivial way of using SVD as unsupervised feature reduction is the following. Given E as set

Proceedings of the EACL 2009 Workshop on GEMS: GEometical Models of Natural Language Semantics, pages 66–73,
Athens, Greece, 31 March 2009.

of training examples represented in a feature space of n features, we can observe it as a matrix, i.e. a sequence of examples $E = (\overrightarrow{e_1}...\overrightarrow{e_m})$. With SVD, the $n \times m$ matrix E can be factorized as $E = U\Sigma V^T$. This factorization implies we can focus the learning problem on a new space using the transformation provided by the matrix U. This new space is represented by the matrix:

$$E' = U^T E = \Sigma V^T \quad (1)$$

where each example is represented with r new features. Each new feature is obtained as a linear combination of the original features, i.e. each feature vector $\overrightarrow{e_l}$ can be seen as a new feature vector $\overrightarrow{e_l}' = U^T \overrightarrow{e_l}$. When the target feature space is big whereas the cardinality of the training set is small, i.e., $n >> m$, the application of SVD results in a reduction of the original feature space as the rank r of the matrix E is $r \leq min(n, m)$.

A more interesting way of using SVD as unsupervised feature selection model is to exploit its approximated computations, i.e. :

$$A \approx A_k = U_{m\times k}\Sigma_{k\times k}V^T_{k\times n}$$

where k is smaller than the rank r. The computation algorithm (Golub and Kahan, 1965) is allowed to stop at a given k different from the real rank r. The property of the singular values, i.e., $\delta_1 \geq \delta_2 \geq ... \geq \delta_r > 0$, guarantees that the first k are bigger than the discarded ones. There is a direct relation between the informativeness of the dimension and the value of the singular value. High singular values correspond to dimensions of the new space where examples have more variability whereas low singular values determine dimensions where examples have a smaller variability (see (Liu, 2007)). These dimensions can not be used as discriminative features in learning algorithms. The possibility of computing the approximated version of the matrix gives a powerful method for feature selection and filtering as we can decide in advance how many features or, better, linear combination of original features we want to use.

As feature selection model, SVD is unsupervised in the sense that the feature selection is done without taking into account the final classes of the training examples. This is not always the case, feature selection models such as those based on Information Gain largely use the final classes of training examples. SVD as feature selection is independent from the classification problem.

3 Probabilistic Taxonomy Learning and SVD feature selection

Recently, Snow et al. (2006) introduced a probabilistic model for learning taxonomies form corpora. This probabilistic formulation exploits the two well known hypotheses: the distributional hypothesis (Harris, 1964) and the exploitation of the lexico-syntactic patterns as in (Robison, 1970; Hearst, 1992). Yet, in this formulation, we can positively and naturally introduce our use of SVD as feature selection model.

In the rest of this section we will firstly introduce the probabilistic model (Sec. 3.1) and, then, we will describe how SVD is used as feature selector in the logistic regression that estimates the probabilities of the model. To describe this part we need to go in depth into the definition of the logistic regression (Sec. 3.2) and the way of estimating the regression coefficients (Sec. 3.3). This will open the possibility of describing how we exploit SVD (Sec. 3.4)

3.1 Probabilistic model

In the probabilistic formulation (Snow et al., 2006), the task of learning taxonomies from a corpus is seen as a probability maximization problem. The taxonomy is seen as a set T of assertions R over pairs $R_{i,j}$. If $R_{i,j}$ is in T, i is a concept and j is one of its generalization (i.e., the direct or the indirect generalization). For example, $R_{dog,animal} \in T$ describes that dog is an $animal$. The main innovation of this probabilistic method is the ability of taking into account in a single probability the information coming from the corpus and an existing taxonomy T.

The main probabilities are then: (1) the prior probability $P(R_{i,j} \in T)$ of an assertion $R_{i,j}$ to belong to the taxonomy T and (2) the posterior probability $P(R_{i,j} \in T|\overrightarrow{e}_{i,j})$ of an assertion $R_{i,j}$ to belong to the taxonomy T given a set of evidences $\overrightarrow{e}_{i,j}$ derived from the corpus. Evidences is a feature vector associated with a pair (i, j). For examples, a feature may describe how many times i and j are seen in patterns like *"i as j"* or *"i is a j"*. These among many other features are indicators of an is-a relation between i and j (see (Hearst, 1992)).

Given a set of evidences E over all the relevant word pairs, in (Snow et al., 2006), the probabilistic taxonomy learning task is defined as the problem of finding the taxonomy $\widehat{T}$ that maximizes the

probability of having the evidences E, i.e.:

$$\widehat{T} = \arg\max_{T} P(E|T)$$

In (Snow et al., 2006), this maximization problem is solved with a local search. What is maximized at each step is the increase of the probability $P(E|T)$ of the taxonomy when the taxonomy changes from T to $T' = T \cup N$ where N are the relations added at each step. This increase of probabilities is defined as multiplicative change $\Delta(N)$ as follows:

$$\Delta(N) = P(E|T')/P(E|T) \quad (2)$$

The main innovation of the model in (Snow et al., 2006) is the possibility of adding at each step the best relation $N = \{R_{i,j}\}$ as well as $N = I(R_{i,j})$ that is $R_{i,j}$ with all the relations by the existing taxonomy. We will then experiment with our feature selection methodology in the two different models:

flat: at each iteration step, a single relation is added, i.e. $\widehat{R}_{i,j} = \arg\max_{R_{i,j}} \Delta(R_{i,j})$

inductive: at each iteration step, a set of relations is added, i.e. $I(\widehat{R}_{i,j})$ where $\widehat{R}_{i,j} = \arg\max_{R_{i,j}} \Delta(I(R_{i,j}))$.

The last important fact is that it is possible to demonstrate that

$$\begin{aligned}\Delta(E_{i,j}) &= k \cdot \frac{P(R_{i,j} \in T|\overrightarrow{e}_{i,j})}{1 - P(R_{i,j} \in T|\overrightarrow{e}_{i,j})} = \\ &= k \cdot odds(R_{i,j})\end{aligned}$$

where k is a constant (see (Snow et al., 2006)) that will be neglected in the maximization process. This last equation gives the possibility of using the logistic regression as it is. In the next sections we will see how SVD and the related feature selection can be used to compute the odds.

3.2 Logistic Regression

Logistic Regression (Cox, 1958) is a particular type of statistical model for relating responses Y to linear combinations of predictor variables X. It is a specific kind of Generalized Linear Model (see (Nelder and Wedderburn, 1972)) where its function is the *logit function* and the independent variable Y is a *binary* or *dicothomic* variable which has a Bernoulli distribution. The dependent variable Y takes value 0 or 1. The probability that Y has value 1 is function of the regressors $x = (1, x_1, ..., x_k)$.

The probabilistic taxonomy learner model introduced in the previous section falls in the category of probabilistic models where the logistic regression can be applied as $R_{i,j} \in T$ is the binary dependent variable and $\overrightarrow{e}_{i,j}$ is the vector of its regressors. In the rest of the section we will see how the *odds*, i.e., the multiplicative change, can be computed.

We start from formally describing the Logistic Regression Model. Given the two stochastic variables Y and X, we can define as p the probability of Y to be 1 given that X=x, i.e.:

$$p = P(Y = 1|X = x)$$

The distribution of the variable Y is a Bernulli distribution, i.e.:

$$Y \sim Bernoulli(p)$$

Given the definition of the $logit(p)$ as:

$$logit(p) = \ln\left(\frac{p}{1-p}\right) \quad (3)$$

and given the fact that Y is a Bernoulli distribution, the logistic regression foresees that the logit is a linear combination of the values of the regressors, i.e.,

$$logit(p) = \beta_0 + \beta_1 x_1 + ... + \beta_k x_k \quad (4)$$

where $\beta_0, \beta_1, ..., \beta_k$ are called *regression coefficients* of the variables $x_1, ..., x_k$ respectively.

Given the regression coefficients, it is possible to compute the probability of a given event where we observe the regressors x to be $Y = 1$ or in our case to belong to the taxonomy. This probability can be computed as follows:

$$p(x) = \frac{\exp(\beta_0 + \beta_1 x_1 + ... + \beta_k x_k)}{1 + \exp(\beta_0 + \beta_1 x_1 + ... + \beta_k x_k)}$$

It is obviously trivial to determine the $odds(R_{i,j})$ related to the multiplicative change of the probabilistic taxonomy model. The $odds$ is the ratio between the positive and the negative event. It is defined as follows:

$$odds(R_{i,j}) = \frac{P(R_{i,j} \in T|\overrightarrow{e}_{i,j})}{1 - P(R_{i,j} \in T|\overrightarrow{e}_{i,j})} \quad (5)$$

Then, it is strictly related with the logit, i.e.:

$$odds(R_{i,j}) = \exp(\beta_0 + \overrightarrow{e}_{i,j}^T \beta) \quad (6)$$

The relationship between the possible values of the probability, odds and logit is show in the Table 1.

Probability	*Odds*	*Logit*
$0 \leq p < 0.5$	$[0, 1)$	$(-\infty, 0]$
$0.5 < p \leq 1$	$[1, \infty)$	$[0, \infty)$

Table 1: Relationship between probability, odds and logit

3.3 Estimating Regression Coefficients

The remaining problem is how to estimate the regression coefficients. This estimation is done using the maximal likelihood estimation to prepare a set of linear equations using the above $logit$ definition and, then, solving a linear problem. This will give us the possibility of introducing the necessity of determining a pseudo-inverse matrix where we will use the singular value decomposition and its natural possibility of performing feature selection. Once we have the regression coefficients, we have the possibility of assigning estimating a probability $P(R_{i,j} \in T|\overrightarrow{e}_{i,j})$ given any configuration of the values of the regressors $\overrightarrow{e}_{i,j}$, i.e., the observed values of the features. For sake of simplicity we will hereafter refer to $\overrightarrow{e}_{i,j}$ as $\overrightarrow{e}_l$.

Let assume we have a multiset O of observations extracted from $Y \times E$ where $Y \in \{0, 1\}$ and we know that some of them are positive observations (i.e., $Y = 1$) and some of them are negative observations (i.e., $Y = 0$).

For each pairs the relative configuration $\overrightarrow{e}_l \in E$ that appeared at least once in O, we can determine using the maximal likelihood estimation $P(Y = 1|\overrightarrow{e}_l)$. Then, from the equation of the logit (Eq. 4), we have a linear equation system, i.e.:

$$\overrightarrow{logit(p)} = Q\beta \tag{7}$$

where Q is a matrix that includes a constant column of 1, necessary for the β_0 of the linear combination of the values of the regression. Moreover it includes the transpose of the evidence matrix, i.e. $E = (\overrightarrow{e}_1 ... \overrightarrow{e}_m)$. Therefore the matrix will be:

$$Q = \begin{pmatrix} 1 & e_{11} & e_{12} & \cdots & e_{1n} \\ 1 & e_{21} & e_{22} & \cdots & e_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e_{m1} & e_{m2} & \cdots & e_{mn} \end{pmatrix}$$

The set of equations in Eq. 7 can be solved using multiple linear regression.

In their general form, the equations of multiple linear regression may be written as (Caron et al., 1988):

$$y = X\beta + \varepsilon$$

where:

- y is a column vector $n \times 1$ that includes the observed values of the dependent variables $Y_1, ..., Y_k$;
- X is a matrix $n \times m$ of the values of the regressors that we have observed;
- β is a column vector $m \times 1$ of the regression coefficients;
- ε is a column vector including the stochastic components that have not been observed and that will not be considered later.

In the case X is a rectangular and singular matrix, the system $y = X\beta$ has not a solution. Yet, it is possible to use the principle of the Least Square Estimation. This principle determines the solution β that minimize the residual norm, i.e.:

$$\widehat{\beta} = \arg\min \|X\beta - y\|^2 \tag{8}$$

This problem can be solved by the **Moore-Penrose pseudoinverse** X^+ (Penrose, 1955). Then, the final equation to determine the β is

$$\widehat{\beta} = X^+ y$$

It is important to remark that if the inverse matrix exist $X^+ = X^{-1}$ and that X^+X and XX^+ are symmetric.

For our case, the following equation is valid:

$$\widehat{\beta} = Q^+ \overrightarrow{logit(p)}$$

3.4 Computing Pseudoinverse Matrix with SVD Analysis

We finally reached the point where it is possible to explain our idea that is naturally using singular value decomposition (SVD) as feature selection in a probabilistic taxonomy learner. In the previous sections we described how the probabilities of the taxonomy learner can be estimated using logistic regressions and we concluded that a way to determine the regression coefficients β is computing the **Moore-Penrose pseudoinverse** Q^+. It is possible to compute the **Moore-Penrose pseudoinverse** using the SVD in the following way (Penrose, 1955). Given an SVD decomposition of the

matrix $Q = U\Sigma V^T$ the pseudo-inverse matrix that minimizes the Eq. 9 is:

$$Q^+ = V\Sigma^+ U^T \quad (9)$$

The diagonal matrix Σ^+ is a matrix $r \times r$ obtained first transposing Σ and then calculating the reciprocals of the singular value of Σ. So the diagonal elements of the Σ^+ are $\frac{1}{\delta_1}, \frac{1}{\delta_2}, ..., , \frac{1}{\delta_r}$.

We have now our opportunity of using SVD as natural feature selector as we can compute different approximations of the pseudo-inverse matrix. As we saw in Sec. 2, the algorithm for computing the singular value decomposition can be stopped a different dimensions. We called k the number of dimensions. As we can obtain different SVD as approximations of the original matrix (Eq. 2), we can define different approximations of :

$$Q^+ \approx Q_k^+ = V_{n\times k}\Sigma^+_{k\times k}U^T_{k\times m}$$

In our experiments we will use different values of k to explore the benefits of SVD as feature selector.

4 Experimental Evaluation

In this section, we want to empirically explore whether our use of SVD feature selection positively affects performances of the probabilistic taxonomy learner. The best way of determining how a taxonomy learner is performing is to see if it can replicate an existing "taxonomy". We will experiment with the attempt of replicating a portion of WordNet (Miller, 1995). In the experiments, we will address two issues: 1) determining to what extent SVD feature selection affect performances of the taxonomy learner; 2) determining if SVD as unsupervised feature selection is better for the task than some simpler model for taxonomy learning. We will explore the effects on both the **flat** and the **inductive** probabilistic taxonomy learner.

The rest of the section is organized as follows. In Sec. 4.1 we will describe the experimental setup in terms of: how we selected the portion of WordNet, the description of the corpus used to extract evidences, a description of the feature space we used, and, finally, the description of a baseline models for taxonomy learning we have used. In Sec. 4.2 we will present the results of the experiments in term of performance.

4.1 Experimental Set-up

To completely define the experiments we need to describe some issues: how we defined the taxonomy to replicate, which corpus we have used to extract evidences for pairs of words, which feature space we used, and, finally, the baseline model we compared our feature selection model against.

As target taxonomy we selected a portion of WordNet[1] (Miller, 1995). Namely, we started from the 44 concrete nouns listed in (McRae et al., 2005) and divided in 3 classes: animal, artifact, and vegetable. For sake of comprehension, this set is described in Tab. 2. For each word w, we selected the synset s_w that is compliant with the class it belongs to. We then obtained a set S of synsets (see Tab. 2). We then expanded the set to S' adding the siblings (i.e., the coordinate terms) for each synset in S. The set S' contains 265 coordinate terms plus the 44 original concrete nouns. For each element in S we collected its hyperonym, obtaining the set H. We then removed from the set H the 4 topmosts: *entity*, *unit*, *object*, and *whole*. The set H contains 77 hyperonyms. For the purpose of the experiments we both derived from the previous sets a taxonomy T and produced a set of negative examples $\overline{T}$. The two sets have been obtained as follows. The taxonomy T is the portion of WordNet implied by $O = H \cup S'$, i.e., T contains all the $(s, h) \in O \times O$ that are in WordNet. On the contrary, $\overline{T}$ contains all the $(s, h) \in O \times O$ that are not in WordNet. We then have 5108 positive pairs in T and 52892 negative pairs in $\overline{T}$.

We then split the set $T \cup \overline{T}$ in two parts, training and testing. As we want to see if it is possible to attach the set S' to the right hyperonym, the split has been done as follows. We randomly divided the set S' in two parts S_{tr} and S_{ts}, respectively, of 70% and 30% of the original S'. We then selected as training T_{tr} all the pairs in T containing a synset in S_{tr} and as testing set T_{ts} those pairs of T containing a synset of S_{ts}. For the probabilistic model, T_{tr} is the initial taxonomy whereas $T_{ts} \cup \overline{T}$ is the unknown set.

As corpus we used the *English Web as Corpus* (ukWaC) (Ferraresi et al., 2008). This is a web extracted corpus of about 2700000 web pages containing more than 2 billion words. The corpus contains documents of different topics such as web, computers, education, public sphere, etc.. It has been largely demonstrated that the web documents

[1] We used the version 3.0

	Concrete nouns	*Clas*	*Sense*		*Concrete nouns*	*Clas*	*Sense*
1	banana	Vegetable	1	23	boat	Artifact	0
2	bottle	Artifact	0	24	bowl	Artifact	0
3	car	Artifact	0	25	cat	Animal	0
4	cherry	Vegetable	2	26	chicken	Animal	1
5	chisel	Artifact	0	27	corn	Vegetable	2
6	cow	Animal	0	28	cup	Artifact	0
7	dog	Animal	0	29	duck	Animal	0
8	eagle	Animal	0	30	elephant	Animal	0
9	hammer	Artifact	1	31	helicopter	Artifact	0
10	kettle	Artifact	0	32	knife	Artifact	0
11	lettuce	Vegetable	2	33	lion	Animal	0
12	motorcycle	Artifact	0	34	mushroom	Vegetable	4
13	onion	Vegetable	2	35	owl	Animal	0
14	peacock	Animal	1	36	pear	Vegetable	0
15	pen	Artifact	0	37	pencil	Artifact	0
16	penguin	Animal	0	38	pig	Animal	0
17	pineapple	Vegetable	1	39	potato	Vegetable	2
18	rocket	Artifact	0	40	scissors	Artifact	0
19	screwdriver	Artifact	0	41	ship	Artifact	0
20	snail	Animal	0	42	spoon	Artifact	0
21	swan	Animal	0	43	telephone	Artifact	1
22	truck	Artifact	0	44	turtle	Animal	1

Table 2: Concrete nouns, Classes and senses selected in WordNet

are good models for natural language (Lapata and Keller, 2004).

As the focus of the paper is the analysis of the effect of the SVD feature selection, we used as feature spaces both n-grams and bag-of-words. Out of the $T \cup \overline{T}$, we selected only those pairs that appeared at a distance of at most 3 tokens. Using these 3 tokens, we generated three spaces: (1) 1-gram that contains monograms, (2) 2-gram that contains monograms and bigrams, and (3) the 3-gram space that contains monograms, bigrams, and trigrams. For the purpose of this experiment, we used a reduced stop list as classical stop words as punctuation, parenthesis, the verb *to be* are very relevant in the context of features for learning a taxonomy.

Finally, we want to describe our *baseline model* for taxonomy learning. This model only contains Heart's patterns (Hearst, 1992) as features. The feature value is the point-wise mutual information. These features are in some sense the best features for the task as these have been manually selected after a process of corpus analysis. These baseline features are included in our 3-gram model. We can then compare our best models with this baseline features in order to see if our SVD feature selection model outperforms manual feature selection.

4.2 Results

In the first set of experiments we want to focus on the issue whether or not performances of the probabilistic taxonomy learner is positively affected by the proposed feature selection model based on the singular value decomposition. We then determined the performance with respect to different values of k. This latter represents the number of surviving dimensions where the pseudo-inverse is computed. Then, it represents the number of features the model adopts. We performed this first set of experiments in the 1-gram feature space. Punctuation has been considered. Figure 1 plots the accuracy of the probabilistic learner with respect to the size of the feature set, i.e. the number k of single values considered for computing the pseudo-inverse matrix. To determine if the effect of the feature selection is preserved during the iteration of the local search algorithm, we report curves at different sizes of the set of added pairs. Curves are

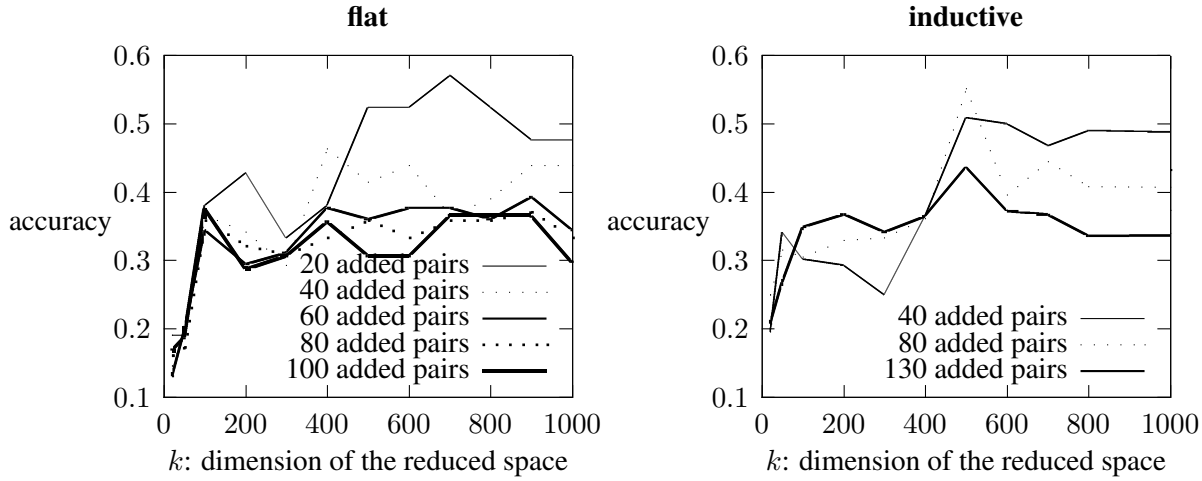


Figure 1: Accuracy over different cuts of the feature space

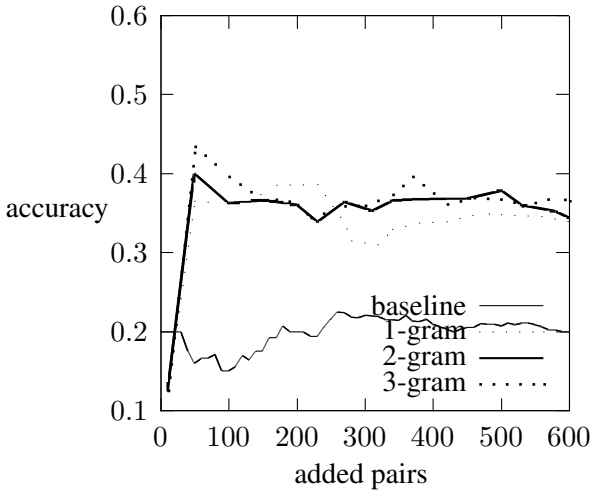


Figure 2: Comparison of different feature spaces with k=400

reported for both the *flat* model and the *inductive* model. The *flat* algorithm adds one pair at each iteration. Then, we reported curves for each 20 added pairs. Each curve shows that accuracy does not increase after a dimension of k=700. This size of the space is necessary only for the first 20 added pairs. Accuracy keeps increasing to k=700 and then decreases. When we add more pairs, the optimal size of the space is around k=200. For the *inductive* model we report the accuracies for around 40, 80, 130 added pairs. Here, at each iteration, more than one pair is added. The optimal dimension of the feature space seems to be around 500 as after that value performances decrease or stay stable. SVD feature selection has then a positive effect for both the *flat* and the *inductive* probabilistic taxonomy learners. This has beneficial effects both on the performances and on the computation time.

In the second set of experiments we want to determine whether or not SVD feature selection for the probabilistic taxonomy learner behaves better than a reduced set of known features. We then fixed the dimension k to 400 and we compared the *baseline model* with different probabilistic models with different feature sets: 1-gram, 2-gram, and 3-gram. We can consider that the trigram model before the cut on its dimensions contains feature subsuming the *baseline model*. Figure 2 shows results. Curves report accuracy after n added pairs. All the probabilistic models outperform the baseline model. As what happened for the first series of experiments (see Fig. 1) more informative spaces such as 3-gram behaves better when the number of

added pairs is small. Performances of the three reduced pairs become similar after 100 added pairs. These experiments show that SVD feature selection has a positive effect on performances as resulting models are always better with respect to the baseline.

5 Conclusions and Future Work

We presented a model to naturally introduce SVD feature selection in a probabilistic taxonomy learner. The method is effective as allows the designing of better probabilistic taxonomy learners. We still need to explore at least two issues. First, we need to determine whether or not the positive effect of SVD feature selection is preserved in more complex feature spaces such as syntactic feature spaces as those used in (Snow et al., 2006). Second, we need to compare the SVD feature selection with other unsupervised feature selection models to determine whether or not this is the best method to use in the case of probabilistic taxonomy learning.

References

D. Caron, W. Hospital, and P. N. Corey. 1988. Variance estimation of linear regression coefficients in complex sampling situation. *Sampling Error: Methodology, Software and Application*, pages 688–694.

D. R. Cox. 1958. The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, 20(2):215–242.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. L, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407.

A. Ferraresi, E. Zanchetta, M. Baroni, and S. Bernardini. 2008. Introducing and evaluating ukwac, a very large web-derived corpus of english. In *In Proceed-ings of the WAC4 Workshop at LREC 2008*, Marrakesh, Morocco.

G. Golub and W. Kahan. 1965. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, 2(2):205–224.

Isabelle Guyon and André Elisseeff. 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, March.

Zellig Harris. 1964. Distributional structure. In Jerrold J. Katz and Jerry A. Fodor, editors, *The Philosophy of Linguistics*, New York. Oxford University Press.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 15th International Conference on Computational Linguistics (CoLing-92)*, Nantes, France.

Mirella Lapata and Frank Keller. 2004. The web as a baseline: Evaluating the performance of unsupervised web-based models for a range of nlp tasks. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, Boston, MA.

Bing Liu. 2007. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Data-Centric Systems and Applications. Springer.

Bernardo Magnini and Manuela Speranza. 2001. Integrating generic and specialized wordnets. In *In Proceedings of the Euroconference RANLP 2001*, Tzigov Chark, Bulgaria.

K. McRae, G.S. Cree, M.S. Seidenberg, and C. McNorgan. 2005. Semantic feature production norms for a large set of living and nonliving things. pages 547–559, Behavioral Research Methods, Instruments, and Computers.

George A. Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41, November.

J. A. Nelder and R. W. M. Wedderburn. 1972. Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3):370–384.

Donie O'Sullivan, A. McElligott, and Richard F. E. Sutcliffe. 1995. Augmenting the princeton wordnet with a domain specific ontology. In *Proceedings of the Workshop on Basic Issues in Knowledge Sharing at the 14th International Joint Conference on Artificial Intelligence*. Montreal, Canada.

Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 113–120, Sydney, Australia, July. Association for Computational Linguistics.

R. Penrose. 1955. A generalized inverse for matrices. In *Proc. Cambridge Philosophical Society*.

Harold R. Robison. 1970. Computer-detectable semantic structures. *Information Storage and Retrieval*, 6(3):273–288.

Rion Snow, Daniel Jurafsky, and A. Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *In ACL*, pages 801–808.

Unsupervised and Constrained Dirichlet Process Mixture Models for Verb Clustering

Andreas Vlachos
Computer Laboratory
University of Cambridge
Cambridge CB3 0FD, UK
av308l@cl.cam.ac.uk

Anna Korhonen
Computer Laboratory
University of Cambridge
Cambridge CB3 0FD, UK
alk23@cl.cam.ac.uk

Zoubin Ghahramani
Department of Engineering
University of Cambridge
Cambridge CB2 1PZ, UK
zoubin@eng.cam.ac.uk

Abstract

In this work, we apply Dirichlet Process Mixture Models (DPMMs) to a learning task in natural language processing (NLP): lexical-semantic verb clustering. We thoroughly evaluate a method of guiding DPMMs towards a particular clustering solution using pairwise constraints. The quantitative and qualitative evaluation performed highlights the benefits of both standard and constrained DPMMs compared to previously used approaches. In addition, it sheds light on the use of evaluation measures and their practical application.

1 Introduction

Bayesian non-parametric models have received a lot of attention in the machine learning community. These models have the attractive property that the number of components used to model the data is not fixed in advance but is actually determined by the model and the data. This property is particularly interesting for NLP where many tasks are aimed at discovering novel, previously unknown information in corpora. Recent work has applied Bayesian non-parametric models to anaphora resolution (Haghighi and Klein, 2007), lexical acquisition (Goldwater, 2007) and language modeling (Teh, 2006) with good results.

Recently, Vlachos et al. (2008) applied the basic models of this class, Dirichlet Process Mixture Models (DPMMs) (Neal, 2000), to a typical learning task in NLP: lexical-semantic verb clustering. The task involves discovering classes of verbs similar in terms of their syntactic-semantic properties (e.g. MOTION class for *travel, walk, run*, etc.). Such classes can provide important support for other NLP tasks, such as word sense disambiguation, parsing and semantic role labeling (Dang, 2004; Swier and Stevenson, 2004). Although some fixed classifications are available (e.g. VerbNet (Kipper-Schuler, 2005)) these are not comprehensive and are inadequate for specific domains (Korhonen et al., 2006b).

Unlike the clustering algorithms applied to this task before, DPMMs do not require the number of clusters as input. This is important because even if the number of classes in a particular task was known (e.g. in the context of a carefully controlled experiment), a particular dataset may not contain instances for all the classes. Moreover, each class is not necessarily contained in one cluster exclusively, since the target classes are defined manually without taking into account the feature representation used. The fact that DPMMs do not require the number of target clusters in advance, renders them promising for the many NLP tasks where clustering is used for learning purposes.

While the results of Vlachos et al. (2008) are promising, the use of a clustering approach which discovers the number of clusters in data presents a new challenge to existing evaluation measures. In this work, we investigate optimal evaluation for such approaches, using the dataset and the basic method of Vlachos et al. as a starting point. We review the applicability of existing evaluation measures and propose a modified version of the newly introduced V-measure (Rosenberg and Hirschberg, 2007). We complement the quantitative evaluation with thorough qualitative assessment, for which we introduce a method to summarize samples obtained from a clustering algorithm.

In preliminary work by Vlachos et al. (2008), a constrained version of DPMMs which takes advantage of *must-link* and *cannot-link* pairwise constraints was introduced. It was demonstrated how such constraines can guide the clustering solution towards some prior intuition or considerations relevant to the specific NLP application in mind. We explain the inference algorithm for the constrained DPMM in greater detail and evaluate quantita-

Proceedings of the EACL 2009 Workshop on GEMS: GEometical Models of Natural Language Semantics, pages 74–82,
Athens, Greece, 31 March 2009.

tively the contribution of each constraint type of independently, complementing it with qualitative analysis. The latter demonstrates how the pairwise constraints added affects instances beyond those involved directly. Finally, we discuss how the unsupervised and the constrained version of DPMMs can be used in a real-world setup.

The results from our comprehensive evaluation show that both versions of DPMMs are capable of learning novel information not in the gold standard, and that the constrained version is more accurate than a previous verb clustering approach which requires setting the number of clusters in advance and is therefore less realistic.

2 Unsupervised clustering with DPMMs

With DPMMs, as with other Bayesian non-parametric models, the number of mixture components is not fixed in advance, but is determined by the model and the data. The parameters of each component are generated by a Dirichlet Process (DP) which can be seen as a distribution over the parameters of other distributions. In turn, each instance is generated by the chosen component given the parameters defined in the previous step:

$$\begin{aligned} G|\alpha, G_0 &\sim DP(\alpha, G_0) \\ \theta_i|G &\sim G \\ x_i|\theta_i &\sim F(\theta_i) \end{aligned} \quad (1)$$

In Eq. 1, G_0 and G are probability distributions over the component parameters (θ), and $\alpha > 0$ is the concentration parameter which determines the variance of the Dirichlet process. We can think of G as a randomly drawn probability distribution with mean G_0. Intuitively, the larger α is, the more similar G will be to G_0. Instance x_i is generated by distribution F, parameterized by θ_i. The graphical model is depicted in Figure 1.

The prior probability of assigning an instance to a particular component is proportionate to the number of instances already assigned to it ($n_{-i,z}$). In other words, DPMMs exhibit the "rich get richer" property. In addition, the probability that a new cluster is created is dependent on the concentration parameter α. A popular metaphor to describe DPMMs which exhibits an equivalent clustering property is the Chinese Restaurant Process (CRP). Customers (instances) arrive at a Chinese restaurant which has an infinite number of tables (components). Each customer sits at one of the tables that is either occupied or vacant with popular tables attracting more customers.

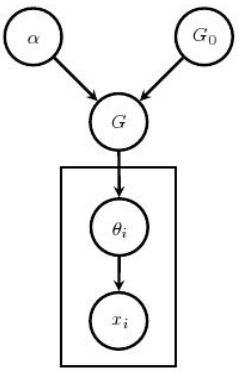


Figure 1: Graphical representation of DPMMs.

In this work, the distribution used to model the components is the multinomial and the prior used is the Dirichlet distribution (F and G_0 in Eq. 1). The conjugacy between them allows for the analytic integration over the component parameters. Following Neal (2000), the component assignments z_i are sampled using the following scheme:

$$\begin{aligned} P(z_i = z|z_{-i}, x_i) \propto \\ p(z_i = z|z_{-i}) DirM(x_i|z_i = z, x_{-i,z}, \lambda) \end{aligned} \quad (2)$$

In Eq. 2 $DirM$ is the Dirichlet-Multinomial distribution, λ are the parameters of the Dirichlet prior G_0 and $x_{-i,z}$ are the instances assigned already to component z (none if we are sampling the probability of assignment to a new component). This sampling scheme is possible due to the fact that the instances in the model are exchangeable, i.e. the order in which they are generated is not relevant.

In terms of the CRP metaphor, we consider each instance x_i as the last customer to arrive and he chooses to sit together with other customers at an existing table or to sit at a new table. Following Navarro et al. (2006) who used the same model to analyze individual differences, we sample the concentration parameter α using the inverse Gamma distribution as a prior.

3 Evaluation measures

The evaluation of unsupervised clustering against a gold standard is not straightforward because the clusters found are not explicitly labelled. Formally defined, an unsupervised clustering algorithm partitions a set of instances $X = \{x_i|i = 1, ..., N\}$ into a set of clusters $K = \{k_j|j = 1, ..., |K|\}$. The standard approach to evaluate the quality of the clusters is to use an external gold standard in which the instances are partitioned into a set of

classes $C = \{c_l | l = 1, ..., |C|\}$. Given this, the goal is to find a partitioning of the instances K that is as close as possible to the gold standard C.

Most work on verb clustering has used the F-measure or the Rand Index (RI) (Rand, 1971) for evaluation, which rely on counting pairwise links between instances. However, Rosenberg and Hirschberg (2007) pointed out that F-measure assumes (the missing) mapping between c_l and k_j. In practice, RI values concentrate in a small interval near 100% (Meilă, 2007).

Rosenberg & Hirschberg (2007) proposed an information-theoretic metric: V-measure. V-measure is the harmonic mean of homogeneity and completeness which evaluate the quality of the clustering in a complementary way. Homogeneity assesses the degree to which each cluster contains instances from a single class of C. This is computed as the conditional entropy of the class distribution of the gold standard given the clustering discovered by the algorithm, $H(C|K)$, normalized by the entropy of the class distribution in the gold standard, $H(C)$. Completeness assesses the degree to which each class is contained in a single cluster. This is computed as the conditional entropy of the cluster distribution discovered by the algorithm given the class, $H(K|C)$, normalized by the entropy of the cluster distribution, $H(K)$. In both cases, we subtract the resulting ratios from 1 to associate higher scores with better solutions:

$$h = 1 - \frac{H(C|K)}{H(C)}$$
$$c = 1 - \frac{H(K|C)}{H(K)}$$
$$V_\beta = \frac{(1+\beta) * h * c}{(\beta * h) + c} \quad (3)$$

The parameter β in Eq. 3 regulates the balance between homogeneity and completeness. Rosenberg & Hirschberg set it to 1 in order to obtain the harmonic mean of these qualities. They also note that V-measure favors clustering solutions with a large number of clusters (large $|K|$), since such solutions can achieve very high homogeneity while maintaining reasonable completeness. This effect is more prominent when a dataset includes a small number of instaces for gold standard classes. While increasing $|K|$ does not guarantee an increase in V-measure (splitting homogeneous clusters would reduce completeness without improving homogeneity), it is easier to achieve higher scores when more clusters are produced.

Another relevant measure is the Variation of Information (VI) (Meilă, 2007). Like V-measure, it assesses homogeneity and completeness using the quantities $H(C|K)$ and $H(K|C)$ respectively, however it simply adds them up to obtain a final result (higher scores are worse). It is also a metric, i.e. VI scores can be added, subtracted, etc, since the quantities involved are measured in bits. However, it can be observed that if $|C|$ and $|K|$ are very different then the terms $H(C|K)$ and $H(K|C)$ will not necessarily be in the same range. In particular, if $|K| \ll |C|$ then $H(K|C)$ (and VI) will be low. In addition, VI scores are not normalized and therefore their interpretation is difficult.

Both V-measure and VI have important advantages over RI and F-measure: they do not assume a mapping between classes and clusters and their scores depend only on the relative sizes of the clusters. However, V-measure and VI can be misleading if the number of clusters found ($|K|$) is substantially different than the number of gold standard classes ($|C|$). In order to ameliorate this, we suggest to take advantage of the β parameter in Eq. 3 in order to balance homogeneity and completeness. More specifically, setting $\beta = |K|/|C|$ assigns more weight to completeness than to homogeneity in case $|K| > |C|$ since the former is harder to achieve and the latter is easier when the clustering solution has more clusters than the gold standard has classes. The opposite occurs when $|K| < |C|$. In case $|K| = |C|$ the score is the same as the original V-measure. Achieving 100% score according to any of these measures requires correct prediction of the number of clusters.

In this work, we evaluate our results using the three measures described above (V-measure, VI, V-beta). We complement this evaluation with qualitative evaluation which assesses the potential of DPMMs to discover novel information that might not be included in the gold standard.

4 Experiments

To perform lexical-semantic verb clustering we used the dataset of Sun et al. (2008). It contains 204 verbs belonging to 17 fine-grained classes in Levin's (1993) taxonomy so that each class contains 12 verbs. The classes and their verbs were selected randomly. The features for each verb are its subcategorization frames (SCFs) and associated frequencies in corpus data, which capture the

	DPMM	Sun et al.
no. of clusters	37.79	17
homogeneity	60.23%	57.57%
completeness	55.82%	60.19%
V-measure	57.94%	58.85%
V-beta	57.11%	58.85%
VI (bits)	3.5746	3.3598

Table 1: Clustering performances.

syntactic context in which the verb occurs. SCFs were extracted from the publicly available VALEX lexicon (Korhonen et al., 2006a). VALEX was acquired automatically using a domain-independent statistical parsing toolkit, RASP (Briscoe and Carroll, 2002), and a classifier which identifies verbal SCFs. As a consequence, it includes some noise due to standard text processing and parsing errors and due to the subtlety of argument-adjunct distinction. In our experiments, we used the SCFs obtained from VALEX1, parameterized for the prepositional frame, which had the best performance in the experiments of Sun et al. (2008).

The feature sets based on verbal SCFs are very sparse and the counts vary over a large range of values. This can be problematic for generative models like DPMMs, since a few dominant features can mislead the model. To reduce the sparsity, we applied non-negative matrix factorization (NMF) (Lin, 2007) which decomposes the dataset in two dense matrices with non-negative values. It has proven useful in a variety of tasks, e.g. information retrieval (Xu et al., 2003) and image processing (Lee and Seung, 1999).

We use a symmetric Dirichlet prior with parameters of 1 (λ in Equation 2). The number of dimensions obtained using NMF was 35. We run the Gibbs sampler 5 times, using 100 iterations for burn-in and draw 20 samples from each run with 5 iterations lag between samples. Table 1 shows the average performances. The DPMM discovers 37.79 verb clusters on average with its performance ranging between 53% and 58% depending on the evaluation measure used. Homogeneity is 4.5% higher than completeness, which is expected since the number of classes in the gold standard is 17. The fact that the DPMM discovers more than twice the number of classes is reflected in the difference between the V-measure and V-beta, the latter being lower. In the same table, we show the results of Sun et al. (2008), who used pairwise clustering (PC) (Puzicha et al., 2000) which involves determining the number of clusters in advance.

The performance of the DPMM is 1%-3% lower than that of Sun et al. As expected, the difference in V-measure is smaller since the DPMM discovers a larger number of clusters, while for VI it is larger. The slightly better performance of PC can be attributed to two factors. First, the (correct) number of clusters is given as input to the PC algorithm and not discovered like by the DPMM. Secondly, PC uses the similarities between the instances to perform the clustering, while the DPMM attempts to find the parameters of the process that generated the data, which is a different and typically a harder task. In addition, the DPMM has two clear advantages which we illustrate in the following sections: it can be used to discover novel information and it can be modified to incorporate intuitive human supervision.

5 Qualitative evaluation

The gold standard employed in this work (Sun et al., 2008) is not fully accurate or comprehensive. It classifies verbs according to their predominant senses in the fairly small SemCor data. Individual classes are relatively coarse-grained in terms of syntactic-semantic analysis[1] and they capture some of the meaning components only. In addition, the gold standard does not capture the semantic relatedness of distinct classes. In fact, the main goal of clustering is to improve such existing classifications with novel information and to create classifications for new domains. We performed qualitative analysis to investigate the extent to which the DPMM meets this goal.

We prepared the data for qualitative analysis as follows: We represented each clustering sample as a linking matrix between the instances of the dataset and measured the frequency of each pair of instances occurring in the same cluster. We constructed a partial clustering of the instances using only those links that occur with frequency higher than a threshold $prob\_link$. Singleton clusters were formed by considering instances that are not linked with any other instances more frequently than a threshold $prob\_single$. The lower the $prob\_link$ threshold, the larger the clusters will be, since more instances get linked. Note that including more links in the solution can either in-

[1] Many original Levin classes have been manually refined in VerbNet.

crease the number of clusters when instances involved were not linked otherwise, or decrease it when linking instances that already belong to other clusters. The higher the $prob\_single$ threshold, the more instances will end up as singletons. By adjusting these two thresholds we can affect the coverage of the analysis. This approach was chosen because it enables to conduct qualitative analysis of data relevant to most clustering samples and irrespective of individual samples. It can also be useful in order to use the output of the clustering algorithm as a component in a pipeline which requires a single result rather than multiple samples.

Using this method, we generated data sets for qualitative analysis using 4 sets of values for $prob\_link$ and $prob\_single$, respectively: (99%, 1%), (95%, 5%), (90%, 10%) and (85%, 15%). Table 1 shows the number of a) verbs, b) clusters (2 or more instances) and c) singletons in each resulting data set, along with the percentage and size of the clusters which represent 1, 2, or multiple gold standard classes. As expected, higher threshold values produce high precision clusters for a smaller set of verbs (e.g. (99%,1%) produces 5 singletons and assigns 70 verbs to 20 clusters, 55% of which represent a single gold standard class), while less extreme threshold values yield higher recall clusters for a larger set of verbs (e.g. (85%,15%) produces 10 singletons and assigns 140 verbs to 25 clusters, 20% of which contain verbs from several gold standard classes).

We conducted the qualitative analysis by comparing the four data sets against the gold standard, SCF distributions, and WordNet (Fellbaum, 1998) senses for each test verb. We first analysed the 5-10 singletons in data sets and discovered that while 3 of the verbs resist classification because of syntactic idiosyncrasy (e.g. *unite* takes intransitive SCFs with frequency higher than other members of class 22.2), the majority of them (7) end up in singletons for valid semantic reasons: taking several frequent WordNet senses they are "too polysemous" to be realistically clustered according to their predominant sense (e.g. *get* and *look*).

We then examined the clusters, and discovered that even in the data set created with the lowest $prob\_link$ threshold of 85%, almost half of the "errors" are in fact novel semantic patterns discovered by clustering. Many of these could be new sub-classes of existing gold standard classes. For example, looking at the 13 high accuracy clusters which correspond to a single gold standard class each, they only represent 9 gold standard classes because as many as 4 classes been divided into two clusters, suggesting that the gold standard is too coarse-grained. Interestingly, each such subdivision seems semantically justified (e.g. the 11.1 PUT verbs *bury* and *immerse* appear in a different cluster than the semantically slightly different *place* and *situate*).

In addition, the DPMM discovers semantically similar gold standard classes. For example, in the data set created with the $prob\_link$ threshold of 99%, 6 of the clusters include members from 2 different gold standard classes. 2 occur due to syntactic idiosyncrasy, but the majority (4) occur because of true semantic relatedness (e.g. the clustering relates 22.2 AMALGAMATE and 36.1 CORRESPOND classes which share similar meaning components). Similarly, in the data set produced by the $prob\_link$ threshold of 85%, one of the largest clusters includes 26 verbs from 5 gold standard classes. The majority of them belong to 3 classes which are related by the meaning component of "motion": 43.1 LIGHT EMISSION, 47.3 MODES OF BEING INVOLVING MOTION, and 51.3.2 RUN verbs:

- **class 22.2** AMALGAMATE: *overlap*
- **class 36.1** CORRESPOND: *banter, concur, dissent, haggle*
- **class 43.1** LIGHT EMISSION: *flare, flicker, gleam, glisten, glow, shine, sparkle*
- **class 47.3** MODES OF BEING INVOLVING MOTION: *falter, flutter, quiver, swirl, wobble*
- **class 51.3.2** RUN: *fly, gallop, glide, jog, march, stroll, swim, travel, trot*

Thus many of the singletons and the clusters in the different outputs capture finer or coarser-grained lexical-semantic differences than those captured in the gold standard. It is encouraging that this happens despite us focussing on a relatively small set of 204 verbs and 17 classes only.

6 Constrained DPMMs

While the ability to discover novel information is attractive in NLP, in many cases it is also desirable to influence the solution with respect to some prior intuition or consideration relevant to the application in mind. For example, while discovering finer-grained classes than those included in the gold standard is useful for some applications, others may benefit from a coarser clustering or a clustering that reveals a specific aspect of the dataset.

THR	verbs	clusters	singletons	% and size of clusters containing		
				1 class	2 classes	multiple classes
99%,1%	70	20	5	55% (3.0)	30% (2.8)	15% (4.5)
95%,5%	104	25	9	40% (3.7)	44% (2.8)	16% (6.8)
90%,10%	128	28	9	46% (3.4)	39% (2.5)	14% (11.0)
85%,15%	140	25	10	44% (3.7)	28% (3.3)	20% (13.0)

Table 2: An overview of the data sets generated for qualitative analysis

Preliminary work by Vlachos et al. (2008) introduced a constrained version of DPMMs that enables human supervision to guide the clustering solution when needed. We model the human supervision as pairwise constraints over instances, following Wagstaff & Cardie (2000): given a pair of instances, they are either linked together (*must-link*) or not (*cannot-link*). For example, *charge* and *run* should form a *must-link* if the aim is to cluster 51.3 MOTION verbs together, but they should form a *cannot-link* if we are interested in 54.5 BILL verbs. In the discussion and the experiments that follow, we assume that all links are consistent with each other. This information can be obtained by asking human experts to label links, or by extracting it from extant lexical resources. Specifying the relations between the instances results in a partial labeling of the instances. Such labeling is likely to be re-usable, since relations between the instances are likely to be useful for a wider range of tasks which might not have identical labels but could still have similar relations.

In order to incorporate the constraints in the DPMM, we modify the underlying generative process to take them into account. In particular *must-linked* instances are generated by the same component and *cannot-linked* instances always by different ones. In terms of the CRP metaphor, customers connected with *must-links* arrive at the restaurant together and choose a table jointly, respecting their *cannot-links* with other customers. They get seated at the same table successively one after the other. Customers without *must-links* with others choose tables avoiding their *cannot-links*.

In order to sample the component assignments according to this model, we restrict the Gibbs sampler to take them into account using the sampling scheme of Fig. 2. First we identify *linked-groups* of instances, taking into account transitivity[2]. We then sample the component assignments only from distributions that respect the links provided. More specifically, for each instance that does not belong to a *linked-group*, we restrict the sampler to choose components that do not contain instances *cannot-linked* with it. For instances in a *linked-group*, we sample their assignment jointly, again taking into account their *cannot-links*. This is performed by adding each instance of the *linked-group* successively to the same component. In Fig. 2, $\mathcal{C}_i$ are the *cannot-links* for instance(s) i, ℓ are the indices of the instances in a *linked-group*, and $z_{<i}$ and $x_{<i}$ are the assignments and the instances of a *linked-group* that have been assigned to a component before instance i.

Input: data $\mathcal{X}$, *must-links* $\mathcal{M}$, *cannot-links* $\mathcal{C}$
$linked\_groups$ = find_linked_groups($\mathcal{X}$, $\mathcal{M}$)
Initialize Z according to $\mathcal{M}$, $\mathcal{C}$
for i **not in** $linked\_groups$
  for $z = 1$ **to** $|Z| + 1$
    if $x_{-i,z} \cap \mathcal{C}_i = \emptyset$
      $P(z_i = z|z_{-i}, x_i)$ (Eq. 2)
    else
      $P(z_i = z|z_{-i}, x_i) = 0$
  Sample from $P(z_i)$
for ℓ **in** $linked\_groups$
  for $z = 1$ **to** $|Z| + 1$
    if $x_{-\ell,z} \cap \mathcal{C}_\ell = \emptyset$
      Set $P(z_\ell = z|z_{-\ell}, x_\ell) = 1$
      for i **in** ℓ
        $P(z_\ell{=}\, z|z_{-\ell}, x_\ell)* =$
          $P(z_i = z|z_{-\ell}, x_{-\ell,z}, z_{<i}, x_{<i})$
    else
      $P(z_\ell = z|z_{-\ell}, x_\ell) = 0$
  Sample from $P(z_\ell)$

Figure 2: Gibbs sampler incorporating *must-links* and *cannot-links*.

7 Experiments using constraints

To investigate the impact of pairwise constraints on clustering by the DPMM, we conduct exper-

[2]If A is linked to B and B to C, then A is linked to C.

iments in which the links are sampled randomly from the gold standard. The number of links varied from 10 to 50 and the random choice was repeated 5 times without checking for redundancy due to transitivity. All the other experimental settings are identical to those in Section 4. Following Wagstaff & Cardie (2000), in Table 3 we show the impact of each link type independently (labeled "must" and "cannot" accordingly), as well as when mixed in equal proportions ("mix").

Adding randomly selected pairwise links is beneficial. In particular, *must-links* improve the clustering rapidly. Incorporating 50 *must-links* improves the performance by 7-8% according to the evaluation measures. In addition, it reduces the average number of clusters by approximately 4. The *cannot-links* are rather ineffective, which is expected as the clustering discovered by the unsupervised DPMM is more fine-grained than the gold standard. For the same reason, it is more likely that the randomly selected *cannot-links* are already discovered by the DPMM and are thus redundant. Wagstaff & Cardie also noted that the impact of the two types of links tends to vary across data sets. Nevertheless, a minor improvement is observed in terms of homogeneity. The balanced mix improves the performance, but less rapidly than the *must-links*.

In order to assess how the links added help the DPMM learn other links we use the Constrained Rand Index (CRI), which is a modification of the Rand Index that takes into account only the pairwise decisions that are not dictated by the constraints added (Wagstaff and Cardie, 2000; Klein et al., 2002). We evaluate the constrained DPMM with CRI (Table 3, bottom right graph) and our results show that the improvements obtained using pairwise constraints are due to learning links beyond the ones enforced.

In a real-world setting, obtaining the mixed set of links is equivalent to asking a human expert to give examples of verbs that should be clustered together or not. Such information could be extracted from a lexical resource (e.g. ontology). Alternatively, the DPMM could be run without any constraints first and if a human expert judges the clustering too coarse (or fine) then *cannot-links* (or *must-links*) could help, since they can adapt the clustering rapidly. When 20 randomly selected *must-links* are integrated, the DPMM reaches or exceeds the performance of PC used by Sun et al. (2008) according to all the evaluation measures. We also argue that it is more realistic to guide the clustering algorithm using pairwise constraints than by defining the number of clusters in advance. Instead of using pairwise constraints to affect the clustering solution, one could alter the parameters for the Dirichlet prior G_0 (Eq. 1) or experiment with varying concentration parameter values. However, it is difficult to predict in advance the exact effect such changes would have in the solution discovered.

Finally, we conducted qualitative analysis of the samples obtained constraining the DPMM with 10 randomly selected *must-links*. We first prepared the data according to the method described in Section 5, using $prob\_link$ and $prob\_single$ thresholds of 99% and 1% respectively. This resulted in 26 clusters and one singleton for 79 verbs. Recall that without constraining the DPMM these thresholds produced 20 clusters and 5 singletons for 70 verbs. 49 verbs are shared in both outputs, while the average cluster size is similar.

The resulting clusters are highly accurate. As many as 16 (i.e. 62%) of them represent a single gold standard class, 7 of which contain (only) the pairs of *must-linked* verbs. Interestingly, only 11 out of 17 gold standard classes are exemplified among the 16 clusters, with 5 classes subdivided into finer-grained classes. Each of these sub-divisions seems semantically fully motivated (e.g. 30.3 PEER verbs were subdivided so that *peep* and *peek* were assigned to a different cluster than the semantically different *gaze*, *glance* and *stare*) and 4 of them can be directly attributed to the use of *must-links*.

From the 6 clusters that contained members from two different gold standard classes, the majority (5) make sense as well. 3 of these contain members of *must-link* pairs together with verbs from semantically related classes (e.g. 37.7 SAY and 40.2 NONVERBAL EXPRESSION classes). 3 of the clusters that contain members of several gold standard classes include *must-link* pairs as well. In two cases *must-links* have helped to bring together verbs which belong to the same class (e.g. the members of the *must-link* pair *broaden-freeze* which represent 45.4 CHANGE OF STATE class appear now in the same cluster with other class members *dampen*, *soften* and *sharpen*). Thus, DPMMs prove useful in learning novel information taking into account pairwise constraints. Only 4

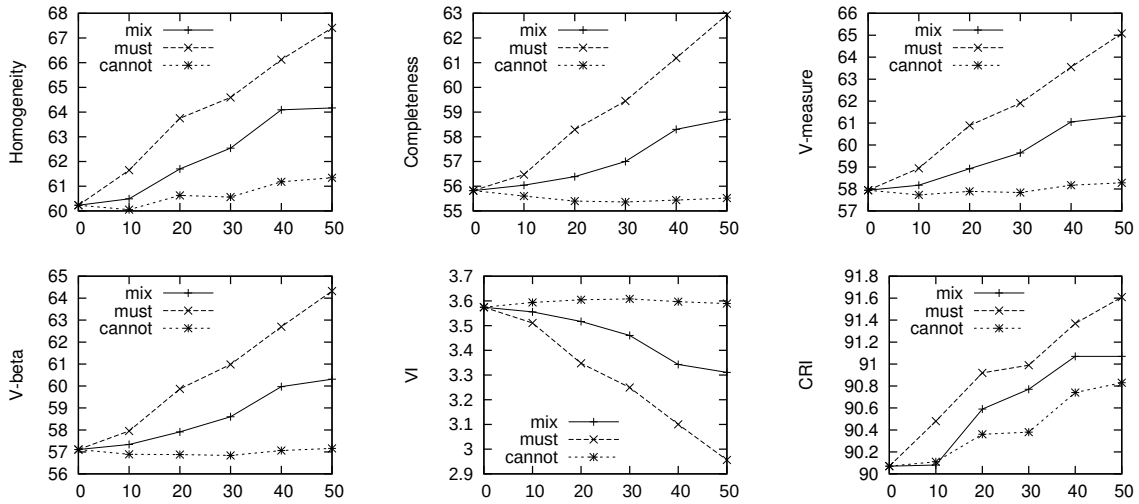


Table 3: Performance of constrained DPMMs incorporating pairwise links.

(i.e. 15%) of the clusters in the output examined are not meaningful (mostly due to the mismatch between the syntax and semantics of verbs).

8 Related work

Previous work on unsupervised verb clustering used algorithms that require the number of clusters as input e.g. PC, Information Bottleneck (Korhonen et al., 2006b) and spectral clustering (Brew and Schulte im Walde, 2002). In terms of applying non-parametric Bayesian approaches to NLP, Haghighi and Klein (2007) evaluated the clustering properties of DPMMs by performing anaphora resolution with good results.

There is a large body of work on semi-supervised learning (SSL), but relatively little work has been done on incorporating some form of supervision in clustering. It is important to note that the pairwise links used in this work constitute a weak form of supervision since they cannot be used to infer class labels which are required for SSL. However, the opposite can be done. Wagstaff & Cardie (2000) employed *must-links* and *cannot-links* to constrain the COBWEB algorithm, while Klein et al. (2002) applied them to complete-link hierarchical agglomerative clustering. The latter also studied how the added links affect instances not directly involved in them.

It can be argued that one could use clustering algorithms that require the number of clusters to be known in advance to discover interesting subclasses such as those discovered by the DPMMs. However, this would normally require multiple runs and manual inspection of the results, while DPMMs discover them automatically. Apart from the fact that fixing the number of clusters in advance restricts the discovery of novel information in the data, such algorithms cannot take full advantage of the pairwise constraints, since the latter are likely to change the number of clusters.

9 Conclusions - Future Work

In this work, following Vlachos et al. (2008) we explored the application of DPMMs to the task of verb clustering. We modified V-measure (Rosenberg and Hirschberg, 2007) to deal more appropriately with the varying number of clusters discovered by DPMMs and presented a method of agregating the generated samples which allows for qualitative evaluation. The quantitative and qualitative evaluation demonstrated that they achieve performance comparable with that of previous work and in addition discover novel information in the data. Furthermore, we evaluated the incorporation of constraints to guide the DPMM obtaining promising results and we discussed their application in a real-world setup.

The results obtained encourage the application of DPMMs and non-parametric Bayesian methods to other NLP tasks. We plan to extend our experiments to larger datasets and further domains. While the improvements achieved using randomly selected pairwise constraints were promising, an active constraint selection scheme as in Klein et al. (2002) could increase their impact. Finally, an extrinsic evaluation of the clustering provided by DPMMs in the context of an NLP application would be informative on their practical potential.

Acknowledgments

We are grateful to Diarmuid Ó Séaghdha and Jurgen Van Gael for helpful discussions.

References

Chris Brew and Sabine Schulte im Walde. 2002. Spectral Clustering for German Verbs. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 117–124.

Ted Briscoe and John Carroll. 2002. Robust accurate statistical annotation of general text. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pages 1499–1504.

Hoa Trang Dang. 2004. *Investigations into the role of lexical semantics in word sense disambiguation*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA, USA.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press.

Sharon J. Goldwater. 2007. *Nonparametric bayesian models of lexical acquisition*. Ph.D. thesis, Brown University, Providence, RI, USA.

Aria Haghighi and Dan Klein. 2007. Unsupervised coreference resolution in a nonparametric bayesian model. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 848–855, Prague, Czech Republic.

Karin Kipper-Schuler. 2005. *VerbNet: A broad-coverage, comprehensive verb lexicon*. Ph.D. thesis, University of Pennsylvania.

Dan Klein, Sepandar Kamvar, and Chris Manning. 2002. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *Proceedings of the Nineteenth International Conference on Machine Learning*.

Anna Korhonen, Yuval Krymolowski, and Ted Briscoe. 2006a. A large subcategorization lexicon for natural language processing applications. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*.

Anna Korhonen, Yuval Krymolowski, and Nigel Collier. 2006b. Automatic classification of verbs in biomedical texts. In *Proceedings of the COLING-ACL*, pages 345–352.

Daniel D. Lee and Sebastian H. Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, October.

Beth Levin. 1993. *English Verb Classes and Alternations: a preliminary investigation*. University of Chicago Press, Chicago.

Chih-Jen Lin. 2007. Projected gradient methods for nonnegative matrix factorization. *Neural Computation*, 19(10):2756–2779.

Marina Meilă. 2007. Comparing clusterings—an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895.

Daniel J. Navarro, Thomas L. Griffiths, Mark Steyvers, and Michael D. Lee. 2006. Modeling individual differences using dirichlet processes. *Journal of Mathematical Psychology*, 50(2):101–122, April.

Radford M. Neal. 2000. Markov Chain Sampling Methods for Dirichlet Process Mixture Models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, June.

Jan Puzicha, Thomas Hofmann, and Joachim Buhmann. 2000. A theory of proximity based clustering: Structure detection by optimization. *Pattern Recognition*, 33(4):617–634.

William M. Rand. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of EMNLP-CoNLL*, pages 410–420, Prague, Czech Republic.

Lin Sun, Anna Korhonen, and Yuval Krymolowski. 2008. Verb class discovery from rich syntactic data. In *Proceedings of the 9th International Conference on Intelligent Text Processing and Computational Linguistics*.

Robert S. Swier and Suzanne Stevenson. 2004. Unsupervised semantic role labelling. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 95–102.

Yee Whye Teh. 2006. A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proceedings of COLING-ACL*, pages 985–992, Sydney, Australia.

Andreas Vlachos, Zoubin Ghahramani, and Anna Korhonen. 2008. Dirichlet process mixture models for verb clustering. In *Proceedings of the ICML workshop on Prior Knowledge for Text and Language*.

Kiri Wagstaff and Claire Cardie. 2000. Clustering with instance-level constraints. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 1103–1110, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Wei Xu, Xin Liu, and Yihong Gong. 2003. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273, New York, NY, USA. ACM Press.

A Non-negative Tensor Factorization Model for Selectional Preference Induction

Tim Van de Cruys
University of Groningen
The Netherlands
t.van.de.cruys@rug.nl

Abstract

Distributional similarity methods have proven to be a valuable tool for the induction of semantic similarity. Up till now, most algorithms use two-way co-occurrence data to compute the meaning of words. Co-occurrence frequencies, however, need not be pairwise. One can easily imagine situations where it is desirable to investigate co-occurrence frequencies of three modes and beyond. This paper will investigate a tensor factorization method called non-negative tensor factorization to build a model of three-way co-occurrences. The approach is applied to the problem of selectional preference induction, and automatically evaluated in a pseudo-disambiguation task. The results show that non-negative tensor factorization is a promising tool for NLP.

1 Introduction

Distributional similarity methods have proven to be a valuable tool for the induction of semantic similarity. The aggregate of a word's contexts generally provides enough information to compute its meaning, viz. its semantic similarity or relatedness to other words.

Up till now, most algorithms use two-way co-occurrence data to compute the meaning of words. A word's meaning might for example be computed by looking at:

- the various documents that the word appears in (words $\times$ documents);
- a bag of words context window around the word (words $\times$ context words);
- the dependency relations that the word appears with (words $\times$ dependency relations).

The extracted data – representing the co-occurrence frequencies of two different entities – is encoded in a matrix. Co-occurrence frequencies, however, need not be pairwise. One can easily imagine situations where it is desirable to investigate co-occurrence frequencies of three modes and beyond. In an information retrieval context, one such situation might be the investigation of *words* $\times$ *documents* $\times$ *authors*. In an NLP context, one might want to investigate *words* $\times$ *dependency relations* $\times$ *bag of word context words*, or *verbs* $\times$ *subjects* $\times$ *direct objects*.

Note that it is not possible to investigate the three-way co-occurrences in a matrix representation form. It is possible to capture the co-occurrence frequencies of a verb with its subjects and its direct objects, but one cannot capture the co-occurrence frequencies of the verb appearing with the subject and the direct object *at the same time*. When the actual three-way co-occurrence data is 'matricized', valuable information is thrown-away. To be able to capture the mutual dependencies among the three modes, we will make use of a generalized *tensor* representation.

Two-way co-occurrence models (such as latent semantic analysis) have often been augmented with some form of dimensionality reduction in order to counter noise and overcome data sparseness. We will also make use of a dimensionality reduction algorithm appropriate for tensor representations.

2 Previous Work

2.1 Selectional Preferences & Verb Clustering

Selectional preferences have been a popular research subject in the NLP community. One of the first to automatically induce selectional preferences from corpora was Resnik (1996). Resnik generalizes among nouns by using WordNet noun

Proceedings of the EACL 2009 Workshop on GEMS: GEometical Models of Natural Language Semantics, pages 83–90,
Athens, Greece, 31 March 2009.

synsets as clusters. He then calculates the *selectional preference strength* of a specific verb in a particular relation by computing the Kullback-Leibler divergence between the cluster distribution of the verb and the aggregate cluster distribution. The *selectional association* is then the contribution of the cluster to the verb's preference strength. The model's generalization relies entirely on WordNet; there is no generalization among the verbs.

The research in this paper is related to previous work on clustering. Pereira et al. (1993) use an information-theoretic based clustering approach, clustering nouns according to their distribution as direct objects among verbs. Their model is a one-sided clustering model: only the direct objects are clustered, there is no clustering among the verbs.

Rooth et al. (1999) use an EM-based clustering technique to induce a clustering based on the co-occurrence frequencies of verbs with their subjects and direct objects. As opposed to the method of Pereira et al. (1993), their model is two-sided: the verbs as well as the subjects/direct objects are clustered. We will use a similar model for evaluation purposes.

Recent approaches using distributional similarity methods for the induction of selectional preferences are the ones by Erk (2007), Bhagat et al. (2007) and Basili et al. (2007).

This research differs from the approaches mentioned above by its use of multi-way data: where the approaches above limit themselves to two-way co-occurrences, this research will focus on co-occurrences for multi-way data.

2.2 Factorization Algorithms

2.2.1 Two-way Factorizations

One of the best known factorization algorithms is principal component analysis (PCA, Pearson (1901)). PCA transforms the data into a new coordinate system, yielding the best possible fit in a least square sense given a limited number of dimensions. Singular value decomposition (SVD) is the generalization of the eigenvalue decomposition used in PCA (Wall et al., 2003).

In information retrieval, singular value decomposition has been applied in latent semantic analysis (LSA, Landauer and Dumais (1997), Landauer et al. (1998)). In LSA, a term-document matrix is created, containing the frequency of each word in a specific document. This matrix is then decomposed into three other matrices with SVD. The most important dimensions that come out of the SVD allegedly represent 'latent semantic dimensions', according to which nouns and documents can be represented more efficiently.

LSA has been criticized for a number of reasons, one of them being the fact that the factorization contains negative numbers. It is not clear what negativity on a semantic scale should designate. Subsequent methods such as probabilistic latent semantic analysis (PLSA, Hofmann (1999)) and non-negative matrix factorization (NMF, Lee and Seung (2000)) remedy these problems, and indeed get much more clear-cut semantic dimensions.

2.2.2 Three-way Factorizations

To be able to cope with three-way data, several algorithms have been developed as multilinear generalizations of the SVD. In statistics, three-way component analysis has been extensively investigated (for an overview, see Kiers and van Mechelen (2001)). The two most popular methods are parallel factor analysis (PARAFAC, Harshman (1970), Carroll and Chang (1970)) and three-mode principal component analysis (3MPCA, Tucker (1966)), also called higher order singular value decomposition (HOSVD, De Lathauwer et al. (2000)). Three-way factorizations have been applied in various domains, such as psychometry and image recognition (Vasilescu and Terzopoulos, 2002). In information retrieval, three-way factorizations have been applied to the problem of link analysis (Kolda and Bader, 2006).

One last important method dealing with multi-way data is non-negative tensor factorization (NTF, Shashua and Hazan (2005)). NTF is a generalization of non-negative matrix factorization, and can be considered an extension of the PARAFAC model with the constraint of non-negativity (cfr. infra).

One of the few papers that has investigated the application of tensor factorization for NLP is Turney (2007), in which a three-mode tensor is used to compute the semantic similarity of words. The method achieves 83.75% accuracy on the TOEFL synonym questions.

3 Methodology

3.1 Tensors

Distributional similarity methods usually represent co-occurrence data in the form of a *matrix*. This form is perfectly suited to represent two-way co-occurrence data, but for co-occurrence data beyond two modes, we need a more general representation. The generalization of a matrix is called a *tensor*. A tensor is able to encode co-occurrence data of any n modes. Figure 1 shows a graphical comparison of a matrix and a tensor with three modes – although a tensor can easily be generalized to more than three modes.

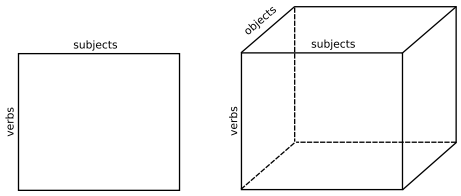


Figure 1: Matrix representation vs. tensor representation

3.2 Non-negative Tensor Factorization

In order to create a succinct and generalized model of the extracted data, a statistical dimensionality reduction technique called non-negative tensor factorization (NTF) is applied to the data. The NTF model is similar to the PARAFAC analysis – popular in areas such as psychology and bio-chemistry – with the constraint that all data needs to be non-negative (i.e. ≥ 0).

Parallel factor analysis (PARAFAC) is a multilinear analogue of the singular value decomposition (SVD) used in latent semantic analysis. The key idea is to minimize the sum of squares between the original tensor and the factorized model of the tensor. For the three mode case of a tensor $T \in \mathbb{R}^{D_1 \times D_2 \times D_3}$ this gives equation 1, where k is the number of dimensions in the factorized model and $\circ$ denotes the outer product.

$$\min_{x_i \in \mathbb{R}^{D1}, y_i \in \mathbb{R}^{D2}, z_i \in \mathbb{R}^{D3}} \| T - \sum_{i=1}^{k} x_i \circ y_i \circ z_i \|_F^2 \qquad (1)$$

With non-negative tensor factorization, the non-negativity constraint is enforced, yielding a model like the one in equation 2:

$$\min_{x_i \in \mathbb{R}^{D1}_{\geq 0}, y_i \in \mathbb{R}^{D2}_{\geq 0}, z_i \in \mathbb{R}^{D3}_{\geq 0}} \| T - \sum_{i=1}^{k} x_i \circ y_i \circ z_i \|_F^2 \qquad (2)$$

The algorithm results in three matrices, indicating the loadings of each mode on the factorized dimensions. The model is represented graphically in figure 2, visualizing the fact that the PARAFAC decomposition consists of the summation over the outer products of n (in this case three) vectors.

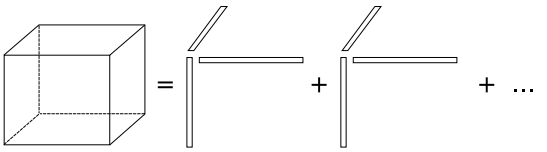


Figure 2: Graphical representation of the NTF as the sum of outer products

Computationally, the non-negative tensor factorization model is fitted by applying an alternating least-squares algorithm. In each iteration, two of the modes are fixed and the third one is fitted in a least squares sense. This process is repeated until convergence.[1]

3.3 Applied to Language Data

The model can straightforwardly be applied to language data. In this part, we describe the factorization of *verbs* $\times$ *subjects* $\times$ *direct objects* co-occurrences, but the example can easily be substituted with other co-occurrence information. Moreover, the model need not be restricted to 3 modes; it is very well possible to go to 4 modes and beyond — as long as the computations remain feasible.

The NTF decomposition for the *verbs* $\times$ *subjects* $\times$ *direct objects* co-occurrences into the three loadings matrices is represented graphically in figure 3. By applying the NTF model to three-way (s, v, o) co-occurrences, we want to extract a generalized selectional preference model, and eventually even induce some kind of frame semantics (in the broad sense of the word).

In the resulting factorization, each verb, subject and direct object gets a loading value for each factor dimension in the corresponding loadings matrix. The original value for a particular (s, v, o)

[1] The algorithm has been implemented in MATLAB, using the Tensor Toolbox for sparse tensor calculations (Bader and Kolda, 2007).

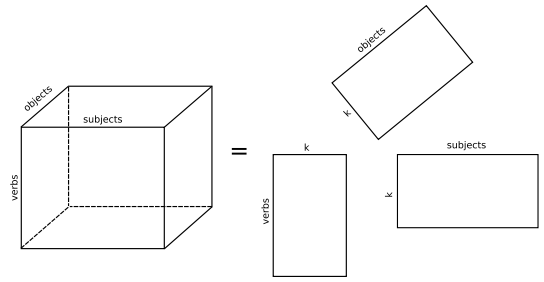


Figure 3: Graphical representation of the NTF for language data

triple x_{svo} can then be reconstructed with equation 3.

$$x_{svo} = \sum_{i=1}^{k} s_{si} v_{vi} o_{oi} \tag{3}$$

To reconstruct the selectional preference value for the triple $(man, bite, dog)$, for example, we look up the subject vector for *man*, the verb vector for *bite* and the direct object vector for *dog*. Then, for each dimension *i* in the model, we multiply the *i*th value of the three vectors. The sum of these values is the final preference value.

4 Results

4.1 Setup

The approach described in the previous section has been applied to Dutch, using the Twente Nieuws Corpus (Ordelman, 2002), a 500M words corpus of Dutch newspaper texts. The corpus has been parsed with the Dutch dependency parser Alpino (van Noord, 2006), and three-way co-occurrences of verbs with their respective subject and direct object relations have been extracted. As dimension sizes, the 1K most frequent verbs were used, together with the 10K most frequent subjects and 10K most frequent direct objects, yielding a tensor of 1K × 10K × 10K. The resulting tensor is very sparse, with only 0.0002% of the values being non-zero.

The tensor has been adapted with a straightforward extension of pointwise mutual information (Church and Hanks, 1990) for three-way co-occurrences, following equation 4. Negative values are set to zero.[2]

$$MI3(x,y,z) = \log \frac{p(x,y,z)}{p(x)p(y)p(z)} \tag{4}$$

The resulting matrix has been factorized into k dimensions (varying between 50 and 300) with the NTF algorithm described in section 3.2.

4.2 Examples

Table 1, 2 and 3 show example dimensions that have been found by the algorithm with $k = 100$. Each example gives the top 10 subjects, verbs and direct objects for a particular dimension, together with the score for that particular dimension. Table 1 shows the induction of a 'police action' frame, with police authorities as subjects, police actions as verbs and patients of the police actions as direct objects.

In table 2, a legislation dimension is induced, with legislative bodies as subjects[3], legislative actions as verbs, and mostly law (proposals) as direct objects. Note that some direct objects (e.g. 'minister') also designate persons that can be the object of a legislative act.

Table 3, finally, is clearly an exhibition dimension, with verbs describing actions of display and trade that art institutions (subjects) can do with works of art (objects).

These are not the only sensible dimensions that have been found by the algorithm. A quick qualitative evaluation indicates that about 44 dimensions contain similar, framelike semantics. In another 43 dimensions, the semantics are less clear-cut (single verbs account for one dimension, or different senses of a verb get mixed up). 13 dimensions are not so much based on semantic characteristics, but rather on syntax (e.g. fixed expressions and pronomina).

4.3 Evaluation

The results of the NTF model have been quantitatively evaluated in a pseudo-disambiguation task, similar to the one used by Rooth et al. (1999). It is used to evaluate the generalization capabilities of the algorithm. The task is to judge which subject (s or s') and direct object (o or o') is more likely for a particular verb v, where (s,v,o) is a combination drawn from the corpus, and s' and o' are a subject and direct object randomly drawn from the corpus. A triple is considered correct if the algorithm prefers both s and o over their counterparts

[2]This is not just an ad hoc conversion to enforce non-negativity. Negative values indicate a smaller co-occurrence probability than the expected number of co-occurrences. Setting those values to zero proves beneficial for similarity calculations (see e.g. Bullinaria and Levy (2007)).

[3]Note that VVD, D66, PvdA and CDA are Dutch political parties.

subjects	su_s	verbs	v_s	objects	obj_s
politie 'police'	.99	*houd_aan* 'arrest'	.64	*verdachte* 'suspect'	.16
agent 'policeman'	.07	*arresteer* 'arrest'	.63	*man* 'man'	.16
autoriteit 'authority'	.05	*pak_op* 'run in'	.41	*betoger* 'demonstrator'	.14
Justitie 'Justice'	.05	*schiet_dood* 'shoot'	.08	*relschopper* 'rioter'	.13
recherche 'detective force'	.04	*verdenk* 'suspect'	.07	*raddraaiers* 'instigator'	.13
marechaussee 'military police'	.04	*tref_aan* 'find'	.06	*overvaller* 'raider'	.13
justitie 'justice'	.04	*achterhaal* 'overtake'	.05	*Roemeen* 'Romanian'	.13
arrestatieteam 'special squad'	.03	*verwijder* 'remove'	.05	*actievoerder* 'campaigner'	.13
leger 'army'	.03	*zoek* 'search'	.04	*hooligan* 'hooligan'	.13
douane 'customs'	.02	*spoor_op* 'track'	.03	*Algerijn* 'Algerian'	.13

Table 1: Top 10 subjects, verbs and direct objects for the 'police action' dimension

subjects	su_s	verbs	v_s	objects	obj_s
meerderheid 'majority'	.33	*steun* 'support'	.83	*motie* 'motion'	.63
VVD	.28	*dien_in* 'submit'	.44	*voorstel* 'proposal'	.53
D66	.25	*neem_aan* 'pass'	.23	*plan* 'plan'	.28
Kamermeerderheid 'Chamber majority'	.25	*wijs_af* 'reject'	.17	*wetsvoorstel* 'bill'	.19
fractie 'party'	.24	*verwerp* 'reject'	.14	*hem* 'him'	.18
PvdA	.23	*vind* 'think'	.08	*kabinet* 'cabinet'	.16
CDA	.23	*aanvaard* 'accepts'	.05	*minister* 'minister'	.16
Tweede Kamer 'Second Chamber'	.21	*behandel* 'treat'	.05	*beleid* 'policy'	.13
partij 'party'	.20	*doe* 'do'	.04	*kandidatuur* 'candidature'	.11
Kamer 'Chamber'	.20	*keur_goed* 'pass'	.03	*amendement* 'amendment'	.09

Table 2: Top 10 subjects, verbs and direct objects for the 'legislation' dimension

s' and o' (so the (s,v,o) triple – that appears in the test corpus – is preferred over the triples (s',v,o'), (s',v,o) and (s,v,o')). Table 4 shows three examples from the pseudo-disambiguation task.

s	v	o	s'	o'
jongere 'youngster'	*drink* 'drink'	*bier* 'beer'	*coalitie* 'coalition'	*aandeel* 'share'
werkgever 'employer'	*riskeer* 'risk'	*boete* 'fine'	*doel* 'goal'	*kopzorg* 'worry'
directeur 'manager'	*zwaai* 'sway'	*scepter* 'sceptre'	*informateur* 'informer'	*vodka* 'wodka'

Table 4: Three examples from the pseudo-disambiguation evaluation task's test set

Four different models have been evaluated. The first two models are tensor factorization models. The first model is the NTF model, as described in section 3.2. The second model is the original PARAFAC model, without the non-negativity constraints.

The other two models are matrix factorization models. The third model is the non-negative matrix factorization (NMF) model, and the fourth model is the singular value decomposition (SVD). For these models, a matrix has been constructed that contains the pairwise co-occurrence frequencies of verbs by subjects as well as direct objects. This gives a matrix of 1K verbs by 10K subjects + 10K direct objects (1K $\times$ 20K). The matrix has been adapted with pointwise mutual information.

The models have been evaluated with 10-fold cross-validation. The corpus contains 298,540 different (s,v,o) co-occurrences. Those have been randomly divided into 10 equal parts. So in each fold, 268,686 co-occurrences have been used for training, and 29,854 have been used for testing. The accuracy results of the evaluation are given in table 5.

The results clearly indicate that the NTF model outperforms all the other models. The model achieves the best result with 300 dimensions, but the differences between the different NTF models are not very large – all attaining scores around 90%.

subjects	su_s	verbs	v_s	objects	obj_s
tentoonstelling 'exhibition'	.50	*toon* 'display'	.72	*schilderij* 'painting'	.47
expositie 'exposition'	.49	*omvat* 'cover'	.63	*werk* 'work'	.46
galerie 'gallery'	.36	*bevat* 'contain'	.18	*tekening* 'drawing'	.36
collectie 'collection'	.29	*presenteer* 'present'	.17	*foto* 'picture'	.33
museum 'museum'	.27	*laat* 'let'	.07	*sculptuur* 'sculpture'	.25
oeuvre 'oeuvre'	.22	*koop* 'buy'	.07	*aquarel* 'aquarelle'	.20
Kunsthal	.19	*bezit* 'own'	.06	*object* 'object'	.19
kunstenaar 'artist'	.15	*zie* 'see'	.05	*beeld* 'statue'	.12
dat 'that'	.12	*koop_aan* 'acquire'	.05	*overzicht* 'overview'	.12
hij 'he'	.10	*in huis heb* 'own'	.04	*portret* 'portrait'	.11

Table 3: Top 10 subjects, verbs and direct objects for the 'exhibition' dimension

	dimensions		
	50 (%)	100 (%)	300 (%)
NTF	**89.52** $\pm$ 0.18	**90.43** $\pm$ 0.14	**90.89** $\pm$ 0.16
PARAFAC	85.57 $\pm$ 0.25	83.58 $\pm$ 0.59	80.12 $\pm$ 0.76
NMF	81.79 $\pm$ 0.15	78.83 $\pm$ 0.40	75.74 $\pm$ 0.63
SVD	69.60 $\pm$ 0.41	62.84 $\pm$ 1.30	45.22 $\pm$ 1.01

Table 5: Results of the 10-fold cross-validation for the NTF, PARAFAC, NMF and SVD model for 50, 100 and 300 dimensions (averages and standard deviation)

The PARAFAC results indicate the fitness of tensor factorization for the induction of three-way selectional preferences. Even without the constraint of non-negativity, the model outperforms the matrix factorization models, reaching a score of about 85%. The model deteriorates when more dimensions are used.

Both matrix factorization models perform worse than their tensor factorization counterparts. The NMF still scores reasonably well, indicating the positive effect of the non-negativity constraint. The simple SVD model performs worst, reaching a score of about 70% with 50 dimensions.

5 Conclusion and Future Work

This paper has presented a novel method that is able to investigate three-way co-occurrences. Other distributional methods deal almost exclusively with pairwise co-occurrences. The ability to keep track of multi-way co-occurrences opens up new possibilities and brings about interesting results. The method uses a factorization model – non-negative tensor factorization – that is suitable for three way data. The model is able to generalize among the data and overcome data sparseness.

The method has been applied to the problem of selectional preference induction. The results indicate that the algorithm is able to induce selectional preferences, leading to a broad kind of frame semantics. The quantitative evaluation shows that use of three-way data is clearly beneficial for the induction of three-way selectional preferences. The tensor models outperform the simple matrix models in the pseudo-disambiguation task. The results also indicate the positive effect of the non-negativity constraint: both models with non-negative constraints outperform their non-constrained counterparts.

The results as well as the evaluation indicate that the method presented here is a promising tool for the investigation of NLP topics, although more research and thorough evaluation are desirable.

There is quite some room for future work. First of all, we want to further investigate the usefulness of the method for selectional preference induction. This includes a deeper quantitative evaluation and a comparison to other methods for selectional preference induction. We also want to include other dependency relations in our model, apart from subjects and direct objects.

Secondly, there is room for improvement and further research with regard to the tensor factorization model. The model presented here minimizes the sum of squared distance. This is, however, not the only objective function possible. Another possibility is the minimization of the Kullback-Leibler divergence. Minimizing the sum of squared distance assumes normally distributed data, and language phenomena are rarely normally distributed. Other objective functions – such as the minimization of the Kullback-Leibler divergence – might be able to capture the language structures

much more adequately. We specifically want to stress this second line of future research as one of the most promising and exciting ones.

Finally, the model presented here is not only suitable for selectional preference induction. There are many problems in NLP that involve three-way co-occurrences. In future work, we want to apply the NTF model presented here to other problems in NLP, the most important one being word sense discrimination.

Acknowledgements

Brett Bader kindly provided his implementation of non-negative tensor factorization for sparse matrices, from which this research has substantially benefited. The three anonymous reviewers provided fruitful comments and remarks, which considerably improved the quality of this paper.

References

Brett W. Bader and Tamara G. Kolda. 2006. Efficient MATLAB computations with sparse and factored tensors. Technical Report SAND2006-7592, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, December.

Brett W. Bader and Tamara G. Kolda. 2007. Matlab tensor toolbox version 2.2. `http://csmr.ca.sandia.gov/~tgkolda/TensorToolbox/`, January.

Roberto Basili, Diego De Cao, Paolo Marocco, and Marco Pennacchiotti. 2007. Learning selectional preferences for entailment or paraphrasing rules. In *Proceedings of RANLP 2007*, Borovets, Bulgaria.

Rahul Bhagat, Patrick Pantel, and Eduard Hovy. 2007. Ledir: An unsupervised algorithm for learning directionality of inference rules. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-07)*, pages 161–170, Prague, Czech Republic.

John A. Bullinaria and Joseph P. Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39:510–526.

J. D. Carroll and J.-J. Chang. 1970. Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. *Psychometrika*, 35:283–319.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information & lexicography. *Computational Linguistics*, 16(1):22–29.

Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. 2000. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278.

Katrin Erk. 2007. A simple, similarity-based model for selectional preferences. In *Proceedings of ACL 2007*, Prague, Czech Republic.

R.A. Harshman. 1970. Foundations of the parafac procedure: models and conditions for an "explanatory" multi-mode factor analysis. In *UCLA Working Papers in Phonetics*, volume 16, pages 1–84, Los Angeles. University of California.

Thomas Hofmann. 1999. Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence, UAI'99*, Stockholm.

H.A.L Kiers and I. van Mechelen. 2001. Three-way component analysis: Principles and illustrative application. *Psychological Methods*, 6:84–110.

Tamara Kolda and Brett Bader. 2006. The TOPHITS model for higher-order web link analysis. In *Workshop on Link Analysis, Counterterrorism and Security*.

Thomas Landauer and Susan Dumais. 1997. A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. *Psychology Review*, 104:211–240.

Thomas Landauer, Peter Foltz, and Darrell Laham. 1998. An Introduction to Latent Semantic Analysis. *Discourse Processes*, 25:295–284.

Daniel D. Lee and H. Sebastian Seung. 2000. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562.

R.J.F. Ordelman. 2002. Twente Nieuws Corpus (TwNC), August. Parlevink Language Technology Group. University of Twente.

K. Pearson. 1901. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572.

Fernando Pereira, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of English words. In *31st Annual Meeting of the ACL*, pages 183–190.

Philip Resnik. 1996. Selectional Constraints: An Information-Theoretic Model and its Computational Realization. *Cognition*, 61:127–159, November.

Mats Rooth, Stefan Riezler, Detlef Prescher, Glenn Carroll, and Franz Beil. 1999. Inducing a semantically annotated lexicon via em-based clustering. In *37th Annual Meeting of the ACL*.

Amnon Shashua and Tamir Hazan. 2005. Non-negative tensor factorization with applications to statistics and computer vision. In *ICML '05: Proceedings of the 22nd international conference on*

Machine learning, pages 792–799, New York, NY, USA. ACM.

L.R. Tucker. 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31:279–311.

Peter D. Turney. 2007. Empirical evaluation of four tensor decomposition algorithms. Technical Report ERB-1152, National Research Council, Institute for Information Technology.

Gertjan van Noord. 2006. At Last Parsing Is Now Operational. In Piet Mertens, Cedrick Fairon, Anne Dister, and Patrick Watrin, editors, *TALN06. Verbum Ex Machina. Actes de la 13e conference sur le traitement automatique des langues naturelles*, pages 20–42, Leuven.

M. Alex O. Vasilescu and Demetri Terzopoulos. 2002. Multilinear analysis of image ensembles: Tensorfaces. In *ECCV*, pages 447–460.

Michael E. Wall, Andreas Rechtsteiner, and Luis M. Rocha, 2003. *Singular Value Decomposition and Principal Component Analysis*, chapter 5, pages 91–109. Kluwel, Norwell, MA, Mar.

A Graph-Theoretic Algorithm for Automatic Extension of Translation Lexicons

Beate Dorow Florian Laws Lukas Michelbacher Christian Scheible Jason Utt
Institute for Natural Language Processing
Universität Stuttgart
{dorowbe,lawsfn,michells,scheibcn,uttjn}@ims.uni-stuttgart.de

Abstract

This paper presents a graph-theoretic approach to the identification of yet-unknown word translations. The proposed algorithm is based on the recursive SimRank algorithm and relies on the intuition that two words are similar if they establish similar grammatical relationships with similar other words. We also present a formulation of SimRank in matrix form and extensions for edge weights, edge labels and multiple graphs.

1 Introduction

This paper describes a cross-linguistic experiment which attempts to extend a given translation dictionary with translations of novel words.

In our experiment, we use an English and a German text corpus and represent each corpus as a graph whose nodes are words and whose edges represent grammatical relationships between words. The corpora need not be parallel.

Our intuition is that a node in the English and a node in the German graph are similar (that is, are likely to be translations of one another), if their neighboring nodes are. Figure 1 shows part of the English and the German word graph.

Many of the (first and higher order) neighbors of *food* and *Lebensmittel* translate to one another (marked by dotted lines), indicating that *food* and *Lebensmittel*, too, are likely mutual translations.

Our hypothesis yields a recursive algorithm for computing node similarities based on the similarities of the nodes they are connected to. We initialize the node similarities using an English-German dictionary whose entries correspond to known pairs of equivalent nodes (words). These node equivalences constitute the "seeds" from which novel English-German node (word) correspondences are bootstrapped.

We are not aware of any previous work using a measure of similarity between nodes in graphs for cross-lingual lexicon acquisition.

Our approach is appealing in that it is language independent, easily implemented and visualized, and readily generalized to other types of data.

Section 2 is dedicated to related research on the automatic extension of translation lexicons. In Section 3 we review SimRank (Jeh and Widom, 2002), an algorithm for computing similarities of nodes in a graph, which forms the basis of our work. We provide a formulation of SimRank in terms of simple matrix operations which allows an efficient implementation using optimized matrix packages. We further present a generalization of SimRank to edge-weighted and edge-labeled graphs and to inter-graph node comparison.

Section 4 describes the process used for building the word graphs. Section 5 presents an experiment for evaluating our approach to bilingual lexicon acquisition. Section 6 reports the results. We present our conclusions and directions for future research in Section 7.

2 Related Work on cross-lingual lexical acquisition

The work by Rapp (1999) is driven by the idea that a word and its translation to another language are likely to co-occur with similar words. Given a German and an English corpus, he computes two word-by-word co-occurrence matrices, one for each language, whose columns span a vector space representing the corresponding corpus.

In order to find the English translation of a German word, he uses a base dictionary to translate all known column labels to English. This yields a new vector representation of the German word in the English vector space. This mapped vector is then compared to all English word vectors, the most similar ones being candidate translations.

Proceedings of the EACL 2009 Workshop on GEMS: GEometical Models of Natural Language Semantics, pages 91–95,
Athens, Greece, 31 March 2009.

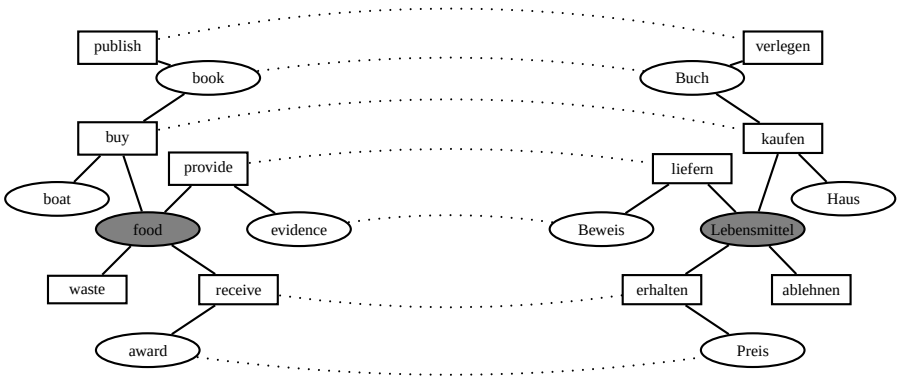


Figure 1: Likely translations based on neighboring nodes

Rapp reports an accuracy of 72% for a small number of test words with well-defined meaning.

Diab and Finch (2000) first compute word similarities within each language corpus separately by comparing their co-occurrence vectors. Their challenge then is to derive a mapping from one language to the other (i.e. a translation lexicon) which best preserves the intra-language word similarities. The mapping is initialized with a few seed "translations" (punctuation marks) which are assumed to be common to both corpora.

They test their method on two corpora written in the same language and report accuracy rates of over 90% on this pseudo-translation task. The approach is attractive in that it does not require a seed lexicon. A drawback is its high computational cost.

Koehn and Knight (2002) use a (linear) combination of clues for bootstrapping an English-German noun translation dictionary. In addition to similar assumptions as above, they consider words to be likely translations of one another if they have the same or similar spelling and/or occur with similar frequencies. Koehn and Knight reach an accuracy of 39% on a test set consisting of the 1,000 most frequent English and German nouns. The experiment excludes verbs whose semantics are more complex than those of nouns.

Otero and Campos (2005) extract English-Spanish pairs of lexico-syntactic patterns from a small parallel corpus. They then construct context vectors for all English and Spanish words by recording their frequency of occurrence in each of these patterns. English and Spanish vectors thus reside in the same vector space and are readily compared.

The approach reaches an accuracy of 89% on a test set consisting of 100 randomly chosen words from among those with a frequency of 100 or higher. The authors do not report results for low-frequency words.

3 The SimRank algorithm

An algorithm for computing similarities of nodes in graphs is the SimRank algorithm (Jeh and Widom, 2002). It was originally proposed for directed unweighted graphs of web pages (nodes) and hyperlinks (links).

The idea of SimRank is to recursively compute node similarity scores based on the scores of neighboring nodes. The similarity S_{ij} of two different nodes i and j in a graph is defined as the normalized sum of the pairwise similarities of their neighbors:

$$S_{ij} = \frac{c}{|N(i)|\ |N(j)|} \sum_{k \in N(i), l \in N(j)} S_{kl}. \qquad (1)$$

$N(i)$ and $N(j)$ are the set of i's and j's neighbors respectively, and c is a multiplicative factor smaller than but close to 1 which demotes the contribution of higher order neighbors. S_{ij} is set to 1 if i and j are identical, which provides a basis for the recursion.

3.1 Matrix formulation of SimRank

We derive a formulation of the SimRank similarity updates which merely consists of matrix multiplications as follows. In terms of the graph's (binary) adjacency matrix A, the SimRank recursion reads:

$$S_{ij} = \frac{c}{|N(i)|\ |N(j)|} \sum_{k \in N(i), l \in N(j)} A_{ik}\, A_{jl}\, S_{kl} \qquad (2)$$

noting that $A_{ik}A_{jl} = 1$, iff k is a neighbor of i and l is a neighbor of j at the same time. This is

equivalent to

$$S_{ij} = c \sum_{k,l} \frac{A_{ik}}{|N(i)|} \frac{A_{jl}}{|N(j)|} S_{kl} \quad (3)$$
$$= c \sum_{k,l} \frac{A_{ik}}{\sum_\nu A_{i\nu}} \frac{A_{jl}}{\sum_\nu A_{j\nu}} S_{kl}.$$

The S_{ij} can be assembled in a square node similarity matrix S, and it is easy to see that the individual similarity updates can be summarized as:

$$S_k = c\, \tilde{A}\, S_{k-1} \tilde{A}^T \quad (4)$$

where $\tilde{A}$ is the row-normalized adjacency matrix and k denotes the current level of recursion. $\tilde{A}$ is obtained by dividing each entry of A by the sum of the entries in its row. The SimRank iteration is initialized with $S = I$, and the diagonal of S, which contains the node self-similarities, is reset to ones after each iteration.

This representation of SimRank in closed matrix form allows the use of optimized off-the-shelf sparse matrix packages for the implementation of the algorithm. This rendered the pruning strategies proposed in the original paper unnecessary. We also note that the Bipartite SimRank algorithm introduced in (Jeh and Widom, 2002) is just a special case of Equation 4.

3.2 Extension with weights and link types

The SimRank algorithm assumes an unweighted graph, i.e. a binary adjacency matrix A. Equation 4 can equally be used to compute similarities in a *weighted* graph by letting $\tilde{A}$ be the graph's row-normalized *weighted* adjacency matrix. The entries of $\tilde{A}$ then represent transition probabilities between nodes rather than hard (binary) adjacency. The proof of the existence and uniqueness of a solution to this more general recursion proceeds in analogy to the proof given in the original paper.

Furthermore, we allow the links in the graph to be of different types and define the following generalized SimRank recursion, where $\mathcal{T}$ is the set of link types and $N_t(i)$ denotes the set of nodes connected to node i via a link of type t.

$$S_{ij} = \frac{c}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \frac{1}{|N_t(i)|\ |N_t(j)|} \sum_{k \in N_t(i), l \in N_t(j)} S_{kl}. \quad (5)$$

In matrix formulation:

$$S_k = \frac{c}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \tilde{A}_t\, S_{k-1} \tilde{A}_t^{\,T} \quad (6)$$

where A_t is the adjacency matrix associated with link type t and, again, may be weighted.

3.3 SimRank across graphs

SimRank was originally designed for the comparison of nodes within a single graph. However, SimRank is readily and accordingly applied to the comparison of nodes of two different graphs. The original SimRank algorithm starts off with the nodes' self-similarities which propagate to other non-identical pairs of nodes. In the case of two different graphs $\mathcal{A}$ and $\mathcal{B}$, we can instead initialize the algorithm with a set of initially known node-node correspondences.

The original SimRank equation (2) then becomes

$$S_{ij} = \frac{c}{|N(i)|\ |N(j)|} \sum_{k,l} A_{ik}\, B_{jl}\, S_{kl}, \quad (7)$$

which is equivalent to

$$S_k = c\, \tilde{A}\, S_{k-1}\, \tilde{B}^T, \quad (8)$$

or, if links are typed,

$$S_k = \frac{c}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \tilde{A}_t\, S_{k-1}\, \tilde{B}_t^{\,T}. \quad (9)$$

The similarity matrix S is now a rectangular matrix containing the similarities between nodes in $\mathcal{A}$ and nodes in $\mathcal{B}$. Those entries of S which correspond to known node-node correspondences are reset to 1 after each iteration.

4 The graph model

The grammatical relationships were extracted from the British National Corpus (BNC) (100 million words), and the Huge German Corpus (HGC) (180 million words of newspaper text). We compiled a list of English verb-object (V-O) pairs based on the verb-argument information extracted by (Schulte im Walde, 1998) from the BNC. The German V-O pairs were extracted from a syntactic analysis of the HGC carried out using the BitPar parser (Schmid, 2004).

We used only V-O pairs because they constitute far more sense-discriminative contexts than, for example, verb-subject pairs, but we plan to examine these and other grammatical relationships in future work.

We reduced English compound nouns to their heads and lemmatized all data. In English phrasal

English						German					
Low		Mid		High		Low		Mid		High	
N	V	N	V	N	V	N	V	N	V	N	V
0.313	0.228	0.253	0.288	0.253	0.255	0.232	0.247	0.205	0.237	0.211	0.205

Table 1: The 12 categories of test words, with mean relative ranks of test words

verbs, we attach the particles to the verbs to distinguish them from the original verb (e.g *put off* vs. *put*). Both the English and German V-O pairs were filtered using stop lists consisting of modal and auxiliary verbs as well as pronouns. To reduce noise, we decided to keep only those relationships which occurred at least three times in the respective corpus.

The English and German data alike are then represented as a bipartite graph whose nodes divide into two sets, verbs and nouns, and whose edges are the V-O relationships which connect verbs to nouns (cf. Figure 1). The edges of the graph are weighted by frequency of occurrence.

We "prune" both the English and German graph by recursively removing all leaf nodes (nodes with a single neighbor). As these correspond to words which appear only in a single relationship, there is only limited evidence of their meaning.

After pruning, there are 4,926 nodes (3,365 nouns, 1,561 verbs) and 43,762 links in the English, and 3,074 nodes (2,207 nouns, 867 verbs) and 15,386 links in the German word graph.

5 Evaluation experiment

The aim of our evaluation experiment is to test the extended SimRank algorithm for its ability to identify novel word translations given the English and German word graph of the previous section and an English-German seed lexicon. We use the dict.cc English-German dictionary [1].

Our evaluation strategy is as follows. We select a set of test words at random from among the words listed in the dictionary, and remove their entries from the dictionary. We run six iterations of SimRank using the remaining dictionary entries as the seed translations (the known node equivalences), and record the similarities of each test word to its known translations. As in the original SimRank paper, c is set to 0.8.

We include both English and German test words and let them vary in frequency: high- (> 100), mid- (> 20 and ≤ 100), and low- (≤ 20) frequent as well as word class (noun, verb). Thus, we obtain 12 categories of test words (summarized in Table 1), each of which is filled with 50 randomly selected words, giving a total of 600 test words.

SimRank returns a matrix of English-German node-node similarities. Given a test word, we extract its row from the similarity matrix and sort the corresponding words by their similarities to the test word. We then scan this sorted list of words and their similarities for the test word's reference translations (those listed in the original dictionary) and record their positions (i.e. ranks) in this list. We then replace absolute ranks with relative ranks by dividing by the total number of candidate translations.

6 Results

Table 1 lists the mean relative rank of the reference translations for each of the test categories. The values of around 0.2-0.3 clearly indicate that our approach ranks the reference translations much higher than a random process would.

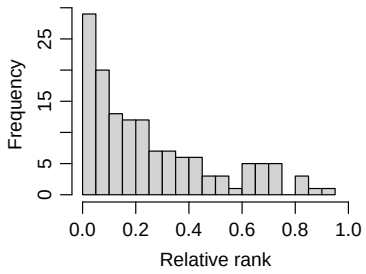


Figure 2: Distribution of the relative ranks of the reference translations in the English-High-N test set.

Exemplary of all test sets, Figure 2 shows the distribution of the relative ranks of the reference translations for the test words in English-High-N. The bulk of the distribution lies below 0.3, i.e. in the top 30% of the candidate list.

In order to give the reader an idea of the results, we present some examples of test words and their

[1] `http://www.dict.cc/` (May 5th 2008)

Test word	Top 10 predicted translations	Ranks
sanction	Ausgangssperre Wirtschaftssanktion Ausnahmezustand Embargo Moratorium *Sanktion* Todesurteil Geldstrafe Bußgeld Anmeldung	Sanktion(6) Maßnahme(1407)
delay	anfechten revidieren zurückstellen füllen verkünden quittieren vertagen verschieben aufheben respektieren	verzögern(78) aufhalten(712)
Kosten	hallmark trouser blouse makup uniform armour robe testimony witness jumper	cost(285)
öffnen	unlock lock usher step peer shut guard hurry slam close	open(12) undo(481)

Table 2: Some examples of test words, their predicted translations, and the ranks of their true translations.

predicted translations in Table 2.

Most of the 10 top-ranked candidate translations of *sanction* are hyponyms of the correct translations. This is mainly due to insufficient noun compound analysis. Both the English and German nouns in our graph model are single words. Whereas the English nouns consist only of head nouns, the German nouns include many compounds (as they are written without spaces), and thus tend to be more specific.

Some of the top candidate translations of *delay* are correct (*verschieben*) or at least acceptable (*vertagen*), but do not count as such as they are missing in the gold standard dictionary.

The mistranslation of the German noun *Kosten* is due to semantic ambiguity. *Kosten* co-occurs often with the verb *tragen* as in *to bear costs*. The verb *tragen* however is ambiguous and may as well be translated as *to wear* which is strongly associated with clothes.

We find several antonyms of *öffnen* among its top predicted translations. Verb-object relationships alone do not suffice to distinguish synonyms from antonyms. Similarly, it is extremely difficult to differentiate between the members of closed categories (e.g. the days of the week, months of the year, mass and time units) using only syntactic relationships.

7 Conclusions and Future Research

The matrix formulation of the SimRank algorithm given in this paper allows an implementation using efficient off-the-shelf software libraries for matrix computation.

We presented an extension of the SimRank algorithm to edge-weighted and edge-labeled graphs. We further generalized the SimRank equations to permit the comparison of nodes from two different graphs, and proposed an application to bilingual lexicon induction.

Our system is not yet accurate enough to be used for actual compilation of translation dictionaries. We further need to address the problem of data sparsity. In particular, we need to remove the bias towards low-degree words whose similarities to other words are unduly high.

In order to solve the problem of ambiguity, we intend to apply SimRank to the incidence representation of the word graphs, which is constructed by putting a node on each link. The proposed algorithm will then naturally return similarities between the more sense-discriminative links (syntactic relationships) in addition to similarities between the often ambiguous nodes (isolated words).

References

M. Diab and S. Finch. 2000. A statistical word-level translation model for comparable corpora. In *In Proceedings of the Conference on Content-Based Multimedia Information Access (RIAO)*.

G. Jeh and J. Widom. 2002. Simrank: A measure of structural-context similarity. In *KDD '02: Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 538–543.

P. Koehn and K. Knight. 2002. Learning a translation lexicon from monolingual corpora. In *Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition*, pages 9–16.

P. Gamallo Otero and J. Ramon Pichel Campos. 2005. An approach to acquire word translations from non-parallel texts. In *EPIA*, pages 600–610.

R. Rapp. 1999. Automatic identification of word translations from unrelated English and German corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 519–526.

Helmut Schmid. 2004. Efficient parsing of highly ambiguous context-free grammars with bit vectors. In *COLING '04: Proceedings of the 20th International Conference on Computational Linguistics*, page 162.

Sabine Schulte im Walde. 1998. Automatic Semantic Classification of Verbs According to Their Alternation Behaviour. Master's thesis, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Handling Sparsity for Verb Noun MWE Token Classification

Mona T. Diab
Center for Computational Learning Systems
Columbia University
mdiab@ccls.columbia.edu

Madhav Krishna
Computer Science Department
Columbia University
madhkrish@gmail.com

Abstract

We address the problem of classifying multiword expression tokens in running text. We focus our study on Verb-Noun Constructions (VNC) that vary in their idiomaticity depending on context. VNC tokens are classified as either idiomatic or literal. Our approach hinges upon the assumption that a literal VNC will have more **in common** with its component words than an idiomatic one. Commonality is measured by contextual overlap. To this end, we set out to explore different contextual variations and different similarity measures handling the sparsity in the possible contexts via four different parameter variations. Our approach yields state of the art performance with an overall accuracy of 75.54% on a TEST data set.

1 Introduction

A Multi-Word Expression (MWE), for our purposes, can be defined as a multi-word unit that refers to a single concept, for example - *kick the bucket*, *spill the beans*, *make a decision*, etc. An MWE typically has an idiosyncratic meaning that is *more* or *different* than the meaning of its component words. An MWE meaning is transparent, i.e. predictable, in as much as the component words in the expression relay the meaning portended by the speaker compositionally. Accordingly, MWEs vary in their degree of meaning compositionality; compositionality is correlated with the level of idiomaticity. An MWE is compositional if the meaning of an MWE as a unit can be predicted from the meaning of its component words such as in *make a decision* meaning *to decide*. If we conceive of idiomaticity as being a continuum, the more idiomatic an expression, the less transparent and the more non-compositional it is.

MWEs are pervasive in natural language, especially in web based texts and speech genres. Identifying MWEs and understanding their meaning is essential to language understanding, hence they are of crucial importance for any Natural Language Processing (NLP) applications that aim at handling robust language meaning and use.

To date, most research has addressed the problem of MWE *type* classification for VNC expressions in English (Melamed, 1997; Lin, 1999; Baldwin et al., 2003; na Villada Moirón and Tiedemann, 2006; Fazly and Stevenson, 2007; Van de Cruys and Villada Moirón, 2007; McCarthy et al., 2007), not *token* classification. For example: *he spilt the beans over the kitchen counter* is most likely a literal usage. This is given away by the use of the prepositional phrase *over the kitchen counter*, since it is plausable that beans could have literally been spilt on a location such as a kitchen counter. Most previous research would classify *spilt the beans* as idiomatic irrespective of usage. A recent study by (Cook et al., 2008) of 60 idiom MWE types concluded that almost half of them had clear literal meaning and over 40% of their usages in text were actually literal. Thus, it would be important for an NLP application such as machine translation, for example, when given a new MWE token, to be able to determine whether it is used idiomatically or not.

In this paper, we address the problem of MWE classification for verb-noun (VNC) token constructions in running text. We investigate the binary classification of an unseen VNC token expression as being either **Idiomatic** (IDM) or **Literal** (LIT). An IDM expression is certainly an MWE, however, the converse is not necessarily true. We handle the problem of *sparsity* for MWE classification by exploring different vector space features: various vector similarity metrics, and more linguistically oriented feature sets. We evaluate our results against a standard data set from the study by (Cook et al., 2007). We achieve state of the art performance in classifying VNC tokens as either literal (F-measure: F_{β_1}=0.64) or idiomatic (F_{β_1}=0.82), corresponding to an overall accuracy of 75.54%.

This paper is organized as follows: In Section

Proceedings of the EACL 2009 Workshop on GEMS: GEometical Models of Natural Language Semantics, pages 96–103,
Athens, Greece, 31 March 2009.

2 we describe our understanding of the various classes of MWEs in general. Section 3 is a summary of previous related research. Section 4 describes our approach. In Section 5 we present the details of our experiments. We discuss the results in Section 6. Finally, we conclude in Section 7.

2 Multi-word Expressions

MWEs are typically not productive, though they allow for inflectional variation (Sag et al., 2002). They have been conventionalized due to persistent use. MWEs can be classified based on their semantic types as follows. **Idiomatic**: This category includes expressions that are semantically non-compositional, *fixed expressions* such as *kingdom come, ad hoc*, *non-fixed expressions* such as *break new ground, speak of the devil*. **Semi-idiomatic**: This class includes expressions that seem semantically non-compositional, yet their semantics are more or less transparent. This category consists of Light Verb Constructions (LVC) such as *make a living* and Verb Particle Constructions (VPC) such as *write-up, call-up*. **Non-Idiomatic**: This category includes expressions that are semantically compositional such as *prime minister*, proper nouns such as *New York Yankees*.

3 Previous Related Work

Several researchers have addressed the problem of MWE classification (Baldwin et al., 2003; Katz and Giesbrecht, 2006; Schone and Juraksfy, 2001; Hashimoto et al., 2006; Hashimoto and Kawahara, 2008). The majority of the proposed research has been using unsupervised approaches and have addressed the problem of MWE type classification irrespective of usage in context. Only, the work by Hashimoto et al. (2006) and Hashimoto and Kawahara (2008) addressed token classification in Japanese using supervised learning.

The most comparable work to ours is the research by (Cook et al., 2007) and (Fazly and Stevenson, 2007). On the other hand, (Cook et al., 2007) develop an unsupervised technique that classifies a VNC expression as idiomatic or literal. They examine if the similarity between the context vector of the MWE, in this case the VNC, and that of its idiomatic usage is higher than the similarity between its context vector and that of its literal usage. They define the vector dimensions in terms of the co-occurrence frequencies of 1000 most frequent content bearing words (nouns, verbs, adjectives, adverbs and determiners) in the corpus. A context vector for a VNC expression is defined in terms of the words in the sentence in which it occurs. They employ the cosine measure to estimate similarity between contextual vectors. They assume that every instance of an expression occurring in a certain *canonical* syntactic form is idiomatic, otherwise it is literal. This assumption holds for many cases of idiomatic usage since many of them are conventionalized, however in cases such as *spilt the beans on the counter top*, the expression would be misclassified as idiomatic since it does occur in the canonical form though the meaning in this case is literal. Their work is similar to this paper in that they explore the VNC expressions at the token level. Their method achieves an accuracy of 52.7% on a data set containing expression tokens used mostly in their literal sense, whereas it yields an accuracy of 82.3% on a data set in which most usages are idiomatic. Further, they report that a classifier that predicts the idiomatic label if an expression (token) occurs in a canonical form achieves an accuracy of 53.4% on the former data set (where the majority of the MWEs occur in their literal sense) and 84.7% on the latter data set (where the majority of the MWE instances are idiomatic). This indicates that these 'canonical' forms can still be used literally. They report an overall system performance accuracy of 72.4%.[1]

(Fazly and Stevenson, 2007) correlate compositionality with idiomaticity. They measure compositionality as a combination of two similarity values: firstly, similar to (Katz and Giesbrecht, 2006), the similarity (cosine similarity) between the context of a VNC and the contexts of its constituent words; secondly, the similarity between an expression's context and that of a verb that is morphologically related to the noun in the expression, for instance, *decide* for *make a decision*. Context $context(t)$ of an expression or a word, t, is defined as a vector of the frequencies of nouns co-occurring with t within a window of ± 5 words. The resulting compositionality measure yields an $F_{\beta=1}$=0.51 on identifying literal expressions and $F_{\beta=1}$=0.42 on identifying idiomatic expressions. However their results are not comparable to ours since it is type-based study.

[1] We note that the use of accuracy as a measure for this work is not the most appropriate since accuracy is a measure of error rather than correctness, hence we report F-measure in addition to accuracy.

4 Our Approach

Recognizing the significance of contextual information in MWE token classification, we explore the space of contextual modeling for the task of classifying the token instances of VNC expressions into literal versus idiomatic expressions. Inspired by works of (Katz and Giesbrecht, 2006; Fazly and Stevenson, 2007), our approach is to compare the context vector of a VNC with the composed vector of the verb and noun (V-N) component units of the VNC when they occur in isolation of each other (i.e., not as a VNC). For example, in the case of the MWE *kick the bucket*, we compare the contexts of the instances of the VNC *kick the bucket* against the combined contexts for the verb (V) *kick*, independent of the noun *bucket*, and the contexts for the noun (N) *bucket*, independent of the verb *kick*. The intuition is that if there is a high similarity between the VNC and the combined V and N (namely, the V-N vector) contexts then the VNC token is compositional, hence a literal instance of the MWE, otherwise the VNC token is idiomatic.

Previous work, (Fazly and Stevenson, 2007), restricted context to within the boundaries of the sentences in which the tokens of interest occurred. We take a cue from that work but define '$context(t)$' as a vector with dimensions as **all** word types occurring in the same sentence as t, where t is a verb type corresponding to the V in the VNC, noun type corresponding to N in the VNC, or VNC expression instance. Moreover, our definition of context includes all nouns, verbs, adjectives and adverbs occurring in the same paragraph as t. This broader notion of context should help reduce sparseness effects, simply by enriching the vector with more contextual information. Further, we realize the importance of some closed class words occurring in the vicinity of t. (Cook et al., 2007) report the importance of determiners in identifying idiomaticity. Prepositions too should be informative of idiomaticity (or literal usage) as illustrated above in *spill the beans on the kitchen counter*. Hence, we include determiners and prepositions occurring in the same sentence as t. The composed V-N contextual vector combines the co-occurrence of the verb type (aggregation of all the verb token instances in the whole corpus) as well as the noun type with this predefined set of dimensions. The VNC contextual vector is that for a specific instance of a VNC expression.

Our objective is to find the best experimental settings that could yield the most accurate classification of VNC expression tokens taking into consideration the sparsity problem. To that end, we explore the space of possible parameter variation on the vectors representing our tokens of interest (VNC, V, or N). We experiment with five different parameter settings:

Context-Extent The definition of context is broad or narrow described as follows. Both $Context_{Broad}$ and $Context_{Narrow}$ comprise all the open class or *content* words (nouns, verbs, adjectives and adverbs), determiners, and prepositions in the **sentence** containing the token. Moreover, $Context_{Broad}$, additionally, includes the content words from the **paragraph** in which the token occurs.

Dimension This is a pruning parameter on the words included from the Context Extent. The intuition is that salient words should have a bigger impact on the calculation of the vector similarity. This parameter is varied in three ways: $Dimension_{NoThresh}$ includes all the words that co-occur with the token under consideration in the specified context extent; $Dimension_{Freq}$ sets a threshold on the co-occurrence frequency for the words to include in the dimensions thereby reducing the dimensionality of the vectors. $Dimension_{Ratio}$ is inspired by the utility of the *tf-idf* measure in information retrieval, we devise a threshold scheme that takes into consideration the salience of the word in context as a function of its relative frequency. Hence the raw frequencies of the words in context are converted to a ratio of two probabilities as per the following equation.

$$ratio = \frac{p(word|context)}{p(word)} = \frac{\frac{freq(word\ in\ context)}{freq(context)}}{\frac{freq(word\ in\ corpus)}{N}} \quad (1)$$

where N is the number of words (tokens) in the corpus and $freq(context)$ is the number of $context$s for a specific token of interest occurs. The numerator of the ratio is the probability that the word occurs in a particular context. The denominator is the probability of occurrence of the word in the corpus. Here, more weight is placed on words that are frequent in a certain context but rarer in the entire corpus. In case of the V and N contexts, a suitable threshold, which is indepen-

dent of data size, is determined on this ratio in order to prune context words.

The latter two pruning techniques, $Dimension_{Freq}$ and $Dimension_{Ratio}$, are not performed for a VNC token's context, hence, all the words in the VNC token's contextual window are included. These thresholding methods are only applied to V-N composed vectors obtained from the combination of the verb and noun vectors.

Context-Content This parameter had two settings: words as they occur in the corpus, $Context - Content_{Words}$; or some of the words are collapsed into named entities, $Context - Content_{Words+NER}$. $Context - Content_{Words+NER}$ attempts to perform dimensionality reduction and sparsity reduction by collapsing named entities. The intuition is that if we reduce the dimensions in semantically salient ways we will not adversely affect performance. We employ BBN's *IdentiFinder* Named Entity Recognition (NER) System[2]. The NER system reduces all proper names, months, days, dates and times to NE tags. NER tagging is done on the corpus before the context vectors are extracted. For our purposes, it is not important that ***John*** *kicked the bucket on* ***Friday*** *in* ***New York City*** – neither the specific actor of the action, nor the place where is occurs is of relevance. The sentence ***PERSON*** *kicked the bucket on* ***DAY*** *in* ***PLACE*** conveys the same amount of information. *IdentiFinder* identifies 24 NE types. We deem 5 of these inaccurate based on our observation, and exclude them. We retain 19 NE types: *Animal, Contact Information, Disease, Event, Facility, Game, Language, Location (merged with Geo-political Entity), Nationality, Organization, Person, Product, Date, Time, Quantity, Cardinal, Money, Ordinal* and *Percentage*. The written-text portion of the BNC contains 6.4M named entities in 5M sentences (at least one NE per sentence). The average number of words per NE is 2.56, the average number of words per sentence is 18.36. Thus, we estimate that by using NER, we reduce vector dimensionality by at least 14% without introducing the negative effects of sparsity.

V-N Combination In order to create a single vector from the units of a VNC expression, we need to combine the vectors pertaining to the verb type (V) and the noun type (N). After combining the word types in the vector dimensions, we need to handle their co-occurrence frequency values. Hence we have two methods: $addition$ where we simply add the frequencies in the cases of the shared dimensions which amounts to a union where the co-occurrence frequencies are added; or $multiplication$ which amounts to an intersection of the vector dimensions where the co-occurrence frequencies are multiplied, hence giving more weight to the shared dimensions than in the $addition$ case. In a study by (Mitchell and Lapata, 2008) on a sentence similarity task, a multiplicative combination model performs better than the additive one.

Similarity Measures We experiment with several standard similarity measures: Cosine Similarity, Overlap similarity, Dice Coefficient and Jaccard Index as defined in (Manning and Schütze, 1999). A context vector is converted to a set by using the dimensions of the vector as members of the set.

5 Experiments and Results

5.1 Data

We use the British National Corpus (BNC),[3] which contains 100M words, because it draws its text from a wide variety of domains and the existing gold standard data sets are derived from it. The BNC contains multiple genres including written text and transcribed speech. We only experiment with the written-text portion. We syntactically parse the corpus with the *Minipar*[4] parser in order to identify all VNC expression tokens in the corpus. We exploit the lemmatized version of the text in order to reduce dimensionality and sparseness. The standard data used in (Cook et al., 2007) (henceforth CFS07) is derived from a set comprising 2920 unique VNC-Token expressions drawn from the whole BNC. In this set, VNC token expressions are manually annotated as *idiomatic*, *literal* or *unknown*.

For our purposes, we discard 127 of the 2920 token gold standard data set either because they are derived from the speech transcription portion of the BNC, or because *Minipar* could not parse them. Similar to the CFS07 set, we exclude expressions labeled *unknown* or pertaining

[2]http://www.bbn.com/technology/identifinder

[3]http://www.natcorp.ox.ac.uk/

[4]http://www.cs.ualberta.ca/ lindek/minipar.htm

to the skewed data set as deemed by the annotators. Therefore, our resulting data set comprises 1125 VNC token expressions (CFS07 has 1180). We then split them into a development (DEV) set and a test (TEST) set. The DEV set comprises 564 token expressions corresponding to 346 idiomatic (IDM) expressions and 218 literal (LIT) ones (CFS07 dev has 573). The TEST set comprises 561 token expressions corresponding to 356 IDM expression tokens and 205 LIT ones (CFS07 test has 607). There is a complete overlap in types between our DEV and CFS07's dev set and our TEST and CFS07's test set. They each comprise 14 VNC type expressions with no overlap in type between the TEST and DEV sets. We divide the tokens between the DEV and TEST maintaining the same proportions of IDM to LIT as recommended in CFS07: DEV is 61.5% and TEST is 63.7%.

5.2 Experimental Set-up

We vary four of the experimental parameters: Context-Extent {sentence only narrow (N), sentence + paragraph broad(B)}, Context-Content {Words (W), Words+NER (NE)}, Dimension {no threshold (nT), frequency (F), ratio (R)}, and V-N compositionality {Additive (A), Multiplicative (M)}. We present the results for all similarity measures. The thresholds (for $Dimension_{Freq}$ and $Dimension_{Ratio}$) are tuned on all the similarity measures collectively. It is observed that the performance of all the measures improved/worsened together, illustrating the same trends in performance, over the various settings of the thresholds evaluated on the DEV data set. Based on tuning on the DEV set, we empirically set the value of the threshold on F to be 188 and for R to be 175 across all experimental conditions. We present results here for 10 experimental conditions based on the four experimental parameters: {**nT-A-W-N, nT-M-W-N, F-A-W-N, F-M-W-N, R-A-W-N, R-M-W-N, R-A-W-B, R-M-W-B, R-A-NE-B, R-M-NE-B**}. For instance, **R-A-W-N**, the Dimension parameter is set to the Ratio $Dimension_{Ratio}$ (R), the V-N compositionality mode is addition (A), and the Context-Content is set to $Context - Content_{Words}$ (W), and, Context-Extent is set to $Context_{Narrow}$ (N).

5.3 Results

We use $F_{\beta=1}$ (F-measure) as the harmonic mean between Precision and Recall, as well as accuracy to report the results. We report the results separately for the two classes IDM and LIT on the DEV and TEST data set for all four similarity measures.

6 Discussion

As shown in Table 2, we obtain the best classification accuracy of 75.54% (R-A-NE-B) on TEST using the Overlap similarity measure, with $F_{\beta=1}$ values for the IDM and LIT classes being 0.82 and 0.64, respectively. These results are generally comparable to state-of-the-art results obtained by CFS07 who report an overall system accuracy of 72.4% on their test set. Hence, we improve over state-of-the-art results by 3% absolute.

In the DEV set, the highest results (F-measures for IDM and LIT, as well as accuracy scores) are obtained for all conditions consistently using the Overlap similarity measure. We also note that our approach tends to fare better overall in classifying IDM than LIT. The best performance is obtained in experimental setting **R-A-NE-B** at 78.53% accuracy corresponding to an IDM classification F-measure of 0.83 and LIT classification F-measure of 0.71.

In the TEST set, we note that Overlap similarity yields the highest overall results, however inconsistently across all the experimental conditions. The highest scores are yielded by the same experimental condition R-A-NE-B. In fact, comparable to previous work, the Cosine similarity measure significantly outperforms the other similarity measures when the Dimension parameter is set to no threshold (nT) and with a set threshold on frequency (F). However, Cosine is outperformed by Overlap when we apply a threshold to the Ratio Dimension. It is worth noting that across all experimental conditions (except in one case, **nT-A-W-N** using Overlap similarity), IDM F-measures are consistently higher than LIT F-measures, suggesting that our approach is more reliable in detecting idiomatic VNC MWE rather than not.

The overall results strongly suggest that using intelligent dimensionality reduction, such as a threshold on the ratio, significantly outperforms no thresholding (nT) and simple frequency thresholding (F) comparing across different similarity measures and all experimental conditions. Recall that R was employed to maintain the salient signals in the context and exclude those that are irrelevant.

Experiment	Dice Coefficient			Jaccard Index			Overlap			Cosine		
	F-measure		Acc. %	F-measure		Acc. %	F-measure		Acc. %	F-measure		Acc. %
	IDM	LIT		IDM	LIT		IDM	LIT		IDM	LIT	
nT-A-W-N	0.45	0.44	44.39	0.47	0.43	44.92	0.50	0.56	**53.30**	0.49	0.42	45.63
nT-M-W-N	0.48	0.46	46.88	0.48	0.46	46.88	0.58	0.57	**57.78**	0.46	0.47	46.52
F-A-W-N	0.47	0.47	46.70	0.47	0.47	46.70	0.58	0.53	**55.62**	0.50	0.50	50.09
F-M-W-N	0.48	0.49	48.31	0.48	0.49	48.31	0.58	0.57	**57.40**	0.54	0.50	52.05
R-A-W-N	0.79	0.62	72.73	0.79	0.62	72.73	0.79	0.63	**73.44**	0.79	0.62	72.73
R-M-W-N	0.76	0.06	62.21	0.76	0.06	62.21	0.77	0.06	**62.39**	0.77	0.06	**62.39**
R-A-W-B	0.59	0.57	58.11	0.59	0.57	58.11	0.80	0.72	**76.47**	0.67	0.65	65.78
R-M-W-B	0.67	0.63	65.06	0.67	0.63	65.06	0.80	0.71	**76.65**	0.71	0.66	68.81
R-A-NE-B	0.58	0.58	58.14	0.58	0.58	58.14	0.83	0.71	**78.53**	0.70	0.64	67.08
R-M-NE-B	0.63	0.63	62.79	0.63	0.63	62.79	0.76	0.69	**73.17**	0.73	0.67	70.13

Table 1: Evaluation on of different experimental conditions on DEV

Experiment	Dice Coefficient			Jaccard Index			Overlap			Cosine		
	F-measure		Acc. %	F-measure		Acc. %	F-measure		Acc. %	F-measure		Acc. %
	IDM	LIT		IDM	LIT		IDM	LIT		IDM	LIT	
nT-A-W-N	0.58	0.48	53.50	0.62	0.49	**56.37**	0.43	0.50	46.32	0.63	0.48	**56.37**
nT-M-W-N	0.58	0.46	52.60	0.53	0.48	50.45	0.53	0.50	51.71	0.55	0.51	**52.78**
F-A-W-N	0.60	0.48	55.12	0.60	0.48	55.12	0.46	0.36	41.47	0.60	0.46	**54.04**
F-M-W-N	0.56	0.48	52.07	0.56	0.48	52.07	0.49	0.45	47.04	0.62	0.49	**56.19**
R-A-W-N	0.81	0.57	73.61	0.81	0.57	73.61	0.82	0.57	**74.51**	0.81	0.57	73.61
R-M-W-N	0.78	0.09	**64.99**	0.78	0.09	**64.99**	0.78	0.08	64.81	0.78	0.08	64.81
R-A-W-B	0.69	0.57	64.11	0.62	0.56	59.11	0.78	0.66	**73.04**	0.68	0.60	64.64
R-M-W-B	0.64	0.60	61.79	0.64	0.60	61.79	0.78	0.64	**72.86**	0.69	0.62	65.89
R-A-NE-B	0.61	0.56	58.45	0.61	0.56	58.45	0.82	0.64	**75.54**	0.68	0.58	63.37
R-M-NE-B	0.59	0.58	58.63	0.59	0.58	58.63	0.76	0.65	**71.40**	0.69	0.61	65.29

Table 2: Evaluation of different experimental conditions on TEST

The results suggest some interaction between the vector combination method, A or M, and the Dimensionality pruning parameters. Experimental conditions that apply the multiplicative compositionality on the component vectors V and N yield higher results in the nT and F conditions across all the similarity measures. Yet once we apply R dimensionality pruning, we see that the additive vector combination, A parameter setting, yields better results. This indicated that the M condition already prunes too much in addition to the R dimensionality hence leading to slightly lower performance.

For both DEV and TEST, we note that the R parameter settings coupled with the A parameter setting. For DEV, we observe that the results yielded from the Broad context extent, contextual sentence and surrounding paragraph, yield higher results than those obtained from the narrow N, context sentence only, across M and A conditions. This trend is not consistent with the results on the TEST data set. R-A-W-N, outperforms R-A-W-B, however, R-M-W-B outperforms R-M-W-N.

We would like to point out that R-M-W-N has very low values for the LIT F-measure, this is attributed to the use of a unified R threshold value of 175. We experimented with different optimal thresholds for R depending on the parameter setting combination and we discovered that for R-M-W-N, the fine-tuned optimal threshold should have been 27 as tuned on the DEV set, yielding LIT F-measures of 0.68 and 0.63, for DEV and TEST, respectively. Hence when using the unified value of 175, more of the compositional vectors components of V+N are pruned away leading to similarity values between the V+N vector and the VNC vector of 0 (across all similarity measures). Accordingly, most of the expressions are

mis-classified as IDM.

The best results overall are yielded from the NE conditions. This result strongly suggests that using class based linguistic information and novel ways to keep the relevant tokens in the vectors such as R yields better MWE classification.

Qualitatively, we note the best results are obtained on the following VNCs from the TEST set in the Overlap similarity measure for the **R-A-W-B** experimental setting (percentage of tokens classified correctly): *make hay(94%), make mark(88%), pull punch (86%), have word(81%), blow whistle (80%), hit wall (79%), hold fire (73%)*. While we note the highest performance on the following VNCs in the corresponding **R-A-NE-B** experimental setting: *make hay(88%), make mark(87%), pull punch (91%), have word(85%), blow whistle (84%), hold fire (82%)*. We observe that both conditions performed the worse on tokens from the following VNCs *lose thread, make hit*. Especially, *make hit* is problematic since it mostly a literal expression, yet in the gold standard set we see it marked inconsistently. For instance, the literal sentence *He bowled it himself and Wilfred Rhodes* **made** *the winning* **hit** while the following annotates *make hit* as idiomatic: *It was the TV show Saturday Night Live which originally* **made** *Martin a huge hit in the States*.

We also note the difference in performance in the hard cases of VNCs that are relatively transparent, only the **R-A-W-B and R-A-NE-B** experimental conditions were able to classify them correctly with high F-measures as either IDM or LIT, namely: *have word, hit wall, make mark*. For **R-A-W-B**, the yielded accuracies are 81%, 79% and 88% respectively, and for **R-A-NE-B**, the accuracies are 85%, 65%, and 87%, respectively. However, in the **nT-A-W-N** condition *have word* is classified incorrectly 82% of the time and in **F-A-W-N** it is classified incorrectly 85% of the time. *Make mark* is classified incorrectly 77% of the time, *make hay* (77%) and *hit wall* (57%) in the **F-A-W-N** experimental setting. This may be attributed to the use of the Broader context, or the use of R in the other more accurate experimental settings.

7 Conclusion

In this study, we explored a set of features that contribute to VNC token expression binary classification. We applied dimensionality reduction heuristics inspired by information retrieval (*tf-idf* like ratio measure) and linguistics (named-entity recogniiton). These contributions improve significantly over experimental conditions that do not manipulate context and dimensions. Our system achieves state-of-the-art performance on a set that is very close to a standard data set. Different from previous studies, we classify VNC token expressions in context. We include function words in modeling the VNC token contexts as well as using the whole paragraph in which it occurs as context. Moreover we empirically show that the Overlap similarity measure is a better measure to use for MWE classification.

8 Acknowledgement

The first author was partially funded by DARPA GALE and MADCAT projects. The authors would like to acknowledge the useful comments by three anonymous reviewers who helped in making this publication more concise and better presented.

References

Timothy Baldwin, Collin Bannard, Takakki Tanaka, and Dominic Widdows. 2003. An empirical model of multiword expression decomposability. In *Proceedings of the ACL 2003 workshop on Multiword expressions*, pages 89–96, Morristown, NJ, USA.

Paul Cook, Afsaneh Fazly, and Suzanne Stevenson. 2007. Pulling their weight: Exploiting syntactic forms for the automatic identification of idiomatic expressions in context. In *Proceedings of the Workshop on A Broader Perspective on Multiword Expressions*, pages 41–48, Prague, Czech Republic, June. Association for Computational Linguistics.

Paul Cook, Afsaneh Fazly, and Suzanne Stevenson. 2008. The VNC-Tokens Dataset. In *Proceedings of the LREC Workshop on Towards a Shared Task for Multiword Expressions (MWE 2008)*, Marrakech, Morocco, June.

Afsaneh Fazly and Suzanne Stevenson. 2007. Distinguishing subtypes of multiword expressions using linguistically-motivated statistical measures. In *Proceedings of the Workshop on A Broader Perspective on Multiword Expressions*, pages 9–16, Prague, Czech Republic, June. Association for Computational Linguistics.

Chikara Hashimoto and Daisuke Kawahara. 2008. Construction of an idiom corpus and its application to idiom identification based on WSD incorporating idiom-specific features. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 992–1001, Hon-

olulu, Hawaii, October. Association for Computational Linguistics.

Chikara Hashimoto, Satoshi Sato, and Takehito Utsuro. 2006. Japanese idiom recognition: Drawing a line between literal and idiomatic meanings. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 353–360, Sydney, Australia, July. Association for Computational Linguistics.

Graham Katz and Eugenie Giesbrecht. 2006. Automatic identification of non-compositional multiword expressions using latent semantic analysis. In *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, pages 12–19, Sydney, Australia, July. Association for Computational Linguistics.

Dekang Lin. 1999. Automatic identification of non-compositional phrases. In *Proceedings of ACL-99*, pages 317–324, Univeristy of Maryland, College Park, Maryland, USA.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press, June.

Diana McCarthy, Sriram Venkatapathy, and Aravind Joshi. 2007. Detecting compositionality of verb-object combinations using selectional preferences. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 369–379, Prague, Czech Republic, June. Association for Computational Linguistics.

Dan I. Melamed. 1997. Automatic discovery of non-compositional compounds in parallel data. In *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing (EMNLP'97)*, pages 97–108, Providence, RI, USA, August.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244, Columbus, Ohio, June. Association for Computational Linguistics.

Bego na Villada Moirón and Jörg Tiedemann. 2006. Identifying idiomatic expressions using automatic word-alignment. In *Proceedings of the EACL-06 Workshop on Multiword Expressions in a Multilingual Context*, pages 33–40, Morristown, NJ, USA.

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann A. Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for nlp. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, pages 1–15, London, UK. Springer-Verlag.

Patrick Schone and Daniel Juraksfy. 2001. Is knowledge-free induction of multiword unit dictionary headwords a solved problem? In *Proceedings of Empirical Methods in Natural Language Processing*, pages 100–108, Pittsburg, PA, USA.

Tim Van de Cruys and Begoña Villada Moirón. 2007. Semantics-based multiword expression extraction. In *Proceedings of the Workshop on A Broader Perspective on Multiword Expressions*, pages 25–32, Prague, Czech Republic, June. Association for Computational Linguistics.

Semantic Density Analysis: Comparing word meaning across time and phonetic space

Eyal Sagi
Northwestern University
Evanston, Illinois, USA
eyal@u.northwestern.edu

Stefan Kaufmann
Northwestern University
Evanston, Illinois, USA
kaufmann@northwestern.edu

Brady Clark
Northwestern University
Evanston, Illinois, USA
bzack@northwestern.edu

Abstract

This paper presents a new statistical method for detecting and tracking changes in word meaning, based on Latent Semantic Analysis. By comparing the density of semantic vector clusters this method allows researchers to make statistical inferences on questions such as whether the meaning of a word changed across time or if a phonetic cluster is associated with a specific meaning. Possible applications of this method are then illustrated in tracing the semantic change of 'dog', 'do', and 'deer' in early English and examining and comparing phonaesthemes.

1 Introduction

The increase in available computing power over the last few decades has led to an explosion in the application of statistical methods to the analysis of texts. Researchers have applied these methods to a wide range of tasks, from word-sense disambiguation (Levin et al., 2006) to the summarization of texts (Marcu, 2003) and the automatic scoring of student essays (Riedel et al., 2006). However, some fields of linguistics that have traditionally employed corpora as their source material, such as historical semantics, have yet to benefit from the application of these statistical methods.

In this paper we demonstrate how an existing statistical tool (Latent Semantic Analysis) can be adapted and used to automate and enhance some aspects of research in historical semantics and other fields whose focus is on the comparative analysis of word meanings within a corpus. Our method allows us to assess the semantic variation within the set of individual occurrences of a given word type. This variation is inversely related to a property of types that we call *density* – intuitively, a tendency to occur in highly similar contexts. In terms of our LSA-based spatial semantic model, we calculate vectors representing the context of each occurrence of a given term, and estimate the term's cohesiveness as the density with which these token context vectors are "packed" in space.

2 The method

Latent Semantic Analysis (LSA) is a collective term for a family of related methods, all of which involve building numerical representations of words based on occurrence patterns in a training corpus. The basic underlying assumption is that co-occurrence within the same contexts can be used as a stand-in measure of semantic relatedness (see Firth, 1957; Halliday and Hasan, 1976; Hoey, 1991, for early articulations of this idea). The success of the method in technical applications such as information retrieval and its popularity as a research tool in psychology, education, linguistics and other disciplines suggest that this hypothesis holds up well for the purposes of those applications.

The relevant notion of "context" varies. The first and still widely used implementation of the idea, developed in Information Retrieval and originally known as Latent Semantic Indexing (Deerwester et al., 1990), assembles a term-document matrix in which each vocabulary item (term) is associated with an n-dimensional vector recording its distribution over the n documents in the corpus. In contrast, the version we applied in this work measures co-occurrence in a way that is more independent of the characteristics of the documents in the training corpus, building in-

Proceedings of the EACL 2009 Workshop on GEMS: GEometical Models of Natural Language Semantics, pages 104–111,
Athens, Greece, 31 March 2009.

stead a term-term matrix associating vocabulary items with vectors representing their frequency of co-occurrence with each of a list of "content-bearing" words. This approach originated with the "WordSpace" paradigm developed by Schütze (1996). The software we used is a version of the "Infomap" package developed at Stanford University and freely available (see also Takayama et al., 1999). We describe it and the steps we took in our experiments in some detail below.

2.1 Word vectors

The information encoded in the co-occurrence matrix, and thus ultimately the similarity measure depends greatly on the genre and subject matter of the training corpus (Takayama et al., 1999; Kaufmann, 2000). In our case, we used the entire available corpus as our training corpus. The word types in the training corpus are ranked by frequency of occurrence, and the Infomap system automatically selects (i) a vocabulary W for which vector representations are to be collected, and (ii) a set C of 1,000 "content-bearing" words whose occurrence or non-occurrence is taken to be indicative of the subject matter of a given passage of text. Usually, these choices are guided by a stoplist of (mostly closed-class) lexical items that are to be excluded, but because we were interested in tracing changes in the meaning of lexical items we reduced this stoplist to a bare minimum. To compensate, we increased the number of "content-bearing" words to 2,000. The vocabulary W consisted of the 40,000 most frequent non-stoplist words. The set C of content-bearing words contained the 50th through 2,049th most frequent non-stoplist words. This method may seem rather blunt, but it has the advantage of not requiring any human intervention or antecedently given information about the domain.

The cells in the resulting matrix of 40,000 rows and 2,000 columns were filled with co-occurrence counts recording, for each pair $\langle w, c\rangle \in W \times C$, the number of times a token of c occurred in the context of a token of w in the corpus.[1] The "context" of a token w_i in our implementation is the set of tokens in a fixed-width window from the 15th item preceding w_i to the 15th item following it (less if a document boundary intervenes). The matrix was transformed by Singular Value Decomposition (SVD), whose implementation in the Infomap system relies on the SVDPACKC package (Berry, 1992; Berry et al., 1993). The output was a reduced 40,000 × 100 matrix. Thus each item $w \in W$ is associated with a 100-dimensional vector $\vec{w}$.

2.2 Context vectors

Once the vector space is obtained from the training corpus, vectors can be calculated for any multi-word unit of text (e.g. paragraphs, queries, or documents), regardless of whether it occurs in the original training corpus or not, as the normalized sum of the vectors associated with the words it contains. In this way, for each occurrence of a target word type under investigation, we calculated a *context vector* from the 15 words preceding and the 15 words following that occurrence.

Context vectors were first used in Word Sense Discrimination by Schütze (1998). Similarly to that application, we assume that these "second-order" vectors encode the aggregate meaning, or topic, of the segment they represent, and thus, following the reasoning behind LSA, are indicative of the meaning with which it is being used on that particular occurrence. Consequently, for each target word of interest, the context vectors associated with its occurrences constitute the data points. The analysis is then a matter of grouping these data points according to some criterion (e.g., the period in which the text was written) and conducting an appropriate statistical test. In some cases it might also be possible to use regression or apply a clustering analysis.

2.3 Semantic Density Analysis

Conducting statistical tests comparing groups of vectors is not trivial. Fortunately, some questions can be answered based on the similarity of vectors within each group rather than the vectors themselves. The similarity between two vectors $\vec{w}$, $\vec{v}$ is measured as the cosine between them:[2]

[1] Two details are glossed over here: First, the Infomap system weighs this raw count with a $tf.idf$ measure of the column label c, calculated as follows: $tf.idf(c) = tf(c) \times \left(log(D+1) - log(df(c))\right)$ where tf and df are the number of occurrences of c and the number of documents in which c occurs, respectively, and D is the total number of documents. Second, the number in each cell is replaced with its square root, in order to approximate a normal distribution of counts and attenuate the potentially distorting influence of high base frequencies (cf. Takayama, et al. 1998; Widdows, 2004).

[2] While the cosine measure is the accepted measure of similarity, the cosine function is non-linear and therefore problematic for many statistical methods. Several transformations can be used to correct this (e.g., Fisher's z). In this paper we will use the angle, in degrees, between the two vectors (i.e., cos^{-1}) because it is easily interpretable.

$$cos(\vec{w}, \vec{v}) = \frac{\vec{w} \cdot \vec{v}}{|\vec{w}||\vec{v}|}$$

The average similarity of a group of vectors is indicative of its density – a dense group of highly similar vectors will have a high average cosine (and a correspondingly low average angle) whereas a sparse group of dissimilar vectors will have an average cosine that approaches zero (and a correspondingly high average angle).[3] Thus since a word that has a single, highly restricted meaning (e.g. 'palindrome') is likely to occur in a very restricted set of contexts, its context vectors are also likely to have a low average angle between them, compared to a word that is highly polysemous or appears in a large variety of contexts (e.g. 'bank', 'do'). From this observation, it follows that it should be possible to compare the cohesiveness of groups of vectors in terms of the average pairwise similarity of the vectors of which they are comprised. Because the number of such pairings tends to be prohibitively large (e.g., nearly 1,000,000 for a group of 1,000 vectors), it is useful to use only a sub-sample in any single analysis. A Monte-Carlo analysis in which n pair-wise similarity values are chosen at random from each group of vectors is therefore appropriate.[4]

However, there is one final complication to consider in the analysis. The passage of time influences not only the meaning of words, but also styles and variety of writing. For example, texts in the 11th century were much less varied, on average, than those written in the 15th century.[5] This will influence the calculation of context vectors as those depend, in part, on the text they are taken from. Because the document as a whole is represented by a vector that is the average of all of its words, it is possible to predict that, if no other factors exist, two contexts are likely to be related to one another to the same degree that their documents are. Controlling for this effect can therefore be achieved by subtracting from the angle between two context vectors the angle between the documents in which they appear.

3 Applications to Research

3.1 A Diachronic Investigation: Semantic Change

One of the central questions of historical semantics is the following (Traugott, 1999):[6]

> Given the form-meaning pair L (lexeme) what changes did meaning M undergo?

For example, the form *as long as* underwent the change `equal in length' > `equal in time' > `provided that'. Evidence for semantic change comes from written records, cognates, and structural analysis (Bloomfield, 1933). Traditional categories of semantic change include (Traugott, 2005: 2-4; Campbell, 2004:254-262; Forston, 2003: 648-650):

- Broadening (generalization, extension, borrowing): A restricted meaning becomes less restricted (e.g. Late Old English *docga* `a (specific) powerful breed of dog' > *dog* `any member of the species *Canis familiaris*'
- Narrowing (specialization, restriction): A relatively general meaning becomes more specific (e.g. Old English *deor* `animal' > *deer*)
- Pejoration (degeneration): A meaning becomes more negative (e.g. Old English *sælig* `blessed, blissful' > *sely* `happy, innocent, pitiable' > *silly* `foolish, stupid')

Semantic change results from the use of language in context, whether linguistic or extralinguistic. Later meanings of forms are connected to earlier ones, where all semantic change arises by polysemy, i.e. new meanings coexist with earlier ones, typically in restricted contexts. Sometimes new meanings split off from earlier ones and are no longer considered variants by language users (e.g. *mistress* `woman in a position of authority, head of household' > `woman in a continuing extra-marital relationship with a man').

Semantic change is often considered unsystematic (Hock and Joseph, 1996: 252). However, recent work (Traugott and Dasher, 2002) suggests that there is, in fact, significant cross-linguistic regularity in semantic change. For ex-

[3] Since the cosine ranges from -1 to +1, it is possible in principle to obtain negative average cosines. In practice, however, the overwhelming majority of vocabulary items have a non-negative cosine with any given target word, hence the average cosine usually does not fall below zero.

[4] It is important to note that the number of *independent samples* in the analysis is determined not by the number of similarity values compared but by the number of individual vectors used in the analysis.

[5] Tracking changes in the distribution of the document vectors in a corpus over time might itself be of interest to some researchers but is beyond the scope of the current paper.

[6] This is the *semasiological* perspective on semantic change. Other perspectives include the *onomasiological* perspective ("Given the concept C, what lexemes can it be expressed by?"). See Traugott 1999 for discussion.

Table 1 - Mean angle between context vectors for target words in different periods in the Helsinki corpus (standard deviations are given in parenthesis)

	n	*Unknown composition date (<1250)*	*Early Middle English (1150-1350)*	*Late Middle English (1350-1500)*	*Early Modern English (1500-1710)*
dog	112			15.47 (14.19)	24.73(10.43)
do	4298		10.31(13.57)	13.02 (9.50)	24.54 (11.2)
deer	61	38.72 (17.59)	20.6 (18.18)		20.5 (9.82)
science	79			13.56 (13.33)	28.31 (12.24)

ample, in the Invited Inferencing Model of Semantic Change proposed by Traugott and Dasher (2002) the main mechanism of semantic change is argued to be the semanticization of *conversational implicatures*, where conversational implicatures are a component of speaker meaning that arises from the interaction between what the speaker says and rational principles of communication (Grice, 1989 [1975]). Conversational implicatures are suggested by an utterance but not entailed. For example, the utterance *Some students came to the party* strongly suggests that some but not all students came to the party, even though the utterance would be true strictly speaking if all students came to the party. According to the Invited Inferencing Model, conversational implicatures become part of the semantic polysemies of particular forms over time.

Such changes in meaning should be evident when examining the contexts in which the lexeme of interest appears. In other words, changes in the meaning of a type should translate to differences in the contexts in which its tokens are used. For instance, semantic broadening results in a meaning that is less restricted and as a result can be used in a larger variety of contexts. In a semantic space that encompasses the period of such a change, this increase in variety can be measured as a decrease in vector density across the time span of the corpus. This decrease translates into an increase in the average angle between the context vectors for the word. For instance, because the Old English word ‘docga’ applied to a specific breed of dog, we predicted that earlier occurrences of the lexemes ‘docga’ and ‘dog’, in a corpus of documents of the appropriate time period, will show less variety than later occurrences.

An even more extreme case of semantic broadening is predicted to occur as part of the process of grammaticalization (Traugot and Dasher, 2002) in which a content word becomes a function word. Because, as a general rule, a function word can be used in a much larger variety of contexts than a content word, a word that underwent grammaticalization should appear in a substantially larger variety of contexts than it did prior to becoming a function word. One well studied case of grammaticalization is that of periphrastic ‘do’. While in Old English ‘do’ was used as a verb with a causative and habitual sense (e.g. ‘do you harm’), later in English it took on a functional role that is nearly devoid of meaning (e.g. ‘do you know him?’). Because this change occurred in Middle English, we predicted that earlier occurrences of ‘do’ will show less variety than later ones.

In contrast with broadening, semantic narrowing results in a meaning that is more restricted, and is therefore applicable in fewer contexts than before. This decrease in variety results in an increase in vector density and can be directly measured as a decrease in the average angle between the context vectors for the word. As an example, the Old English word ‘deor’ denoted a larger group of living creatures than does the Modern English word ‘deer’. We therefore predicted that earlier occurrences of the lexemes ‘deor’ and ‘deer’, in a corpus of the appropriate time period, will show more variety than later occurrences.

We tested our predictions using a corpus derived from the Helsinki corpus (Rissanen, 1994). The Helsinki corpus is comprised of texts spanning the periods of Old English (prior to 1150A.D.), Middle English (1150-1500A.D.), and Early Modern English (1500-1710A.D.). Because spelling in Old English was highly variable, we decided to exclude that part of the corpus and focused our analysis on the Middle English and Early Modern English periods. The resulting corpus included 504 distinct documents totaling approximately 1.1 million words.

To test our predictions regarding semantic change in the words ‘dog’, ‘do’, and ‘deer’, we collected all of the contexts in which they appear in our subset of the Helsinki corpus. This resulted in 112 contexts for ‘dog’, 4298 contexts for ‘do’, and 61 contexts for ‘deer’. Because there were relatively few occurrences of ‘dog’

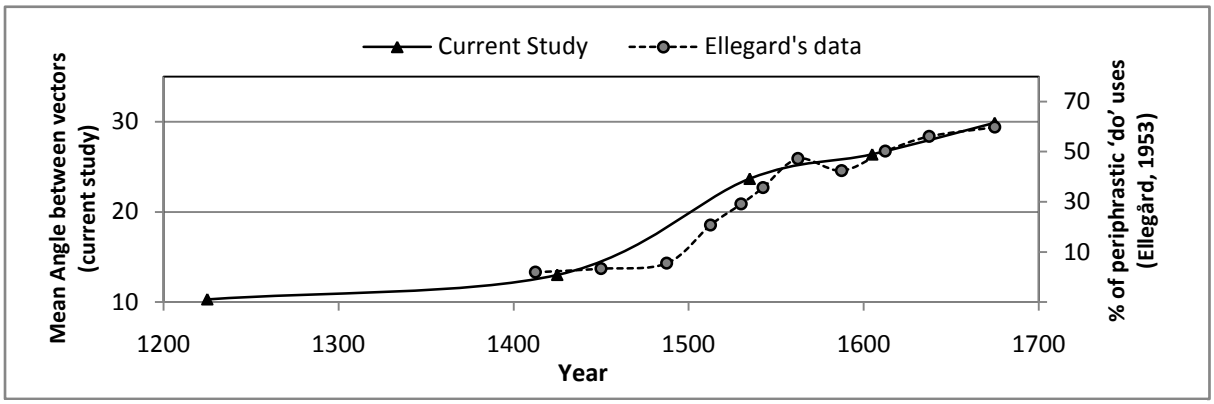


Figure 1 – A comparison of the rise of periphrastic 'do' as measured by semantic density in our study and the proportion of periphrastic 'do' uses by Ellegård (1953).

and 'deer' in the corpus it was practical to compute the angles between all possible pairs of context vectors. As a result, we elected to forgo the Monte-Carlo analysis for those two words in favor of a full analysis. The results of our analysis for all three words are given in Table 1. These results were congruent with our prediction: The density of the contexts decreases over time for both 'dog' ($t(110) = 2.17$, $p < .05$) and 'do' ($F(2,2997)=409.41$, $p < .01$) while in the case of 'deer' there is an increase in the density of the contexts over time ($t(36) = 3.05$, $p < .01$).

Furthermore, our analysis corresponds with the data collected by Ellegård (1953). Ellegård traced the grammaticalization of 'do' by manually examining changes in the proportions of its various uses between 1400 and 1700. His data identifies an overall shift in the pattern of use that occurred mainly between 1475 and 1575. Our analysis identifies a similar shift in patterns between the time periods spanning 1350-1500 and 1500-1570. Figure 1 depicts an overlay of both datasets. The relative scale of the two sets was set so that the proportions of 'do' uses at 1400 and 1700 (the beginning and end of Ellegård's data, respectively) match the semantic density measured by our method at those times.

Finally, our method can be used not only to test predictions based on established cases of semantic change, but also to identify new ones. For instance, in examining the contexts of the word 'science' we can identify that it underwent semantic broadening shortly after it first appeared in the 14th century ($t(77) = 4.51$, $p < .01$). A subsequent examination of the contexts in which the word appears indicated that this is probably the result of a shift from a meaning related to generalized knowledge (e.g., '…and learn science of school', John of Trevisa's Polychronicon, 1387) to one that can also be used to refer to more specific disciplines (e.g., '…of the seven liberal sciences', Simon Forman's Diary, 1602).

Our long term goal with respect to this type of analysis is to use this method in a computer-based tool that can scan a diachronic corpus and automatically identify probable cases of semantic change within it. Researchers can then use these results to focus on identifying the specifics of such changes, as well as examine the overall patterns of change that exist in the corpus. It is our belief that such a use will enable a more rigorous testing and refinement of existing theories of semantic change.

3.2 A Synchronic Investigation: Phonaesthemes

In addition to examining changes in meaning across time, it is also possible to employ our method to examine how the semantic space relates to other possible partitioning of the lexemes represented by it. For instance, while the relationship between the phonetic representation and semantic content is largely considered to be arbitrary, there are some notable exceptions. One interesting case is that of *phonaesthemes* (Firth, 1930), sub-morphemic units that have a predictable effect on the meaning of the word as a whole. In English, one of the more frequently mentioned phonaesthemes is a word-initial *gl-* which is common in words related to the visual modality (e.g., 'glance', 'gleam'). While there have been some scholastic explorations of these non-morphological relationships between sound and meaning, they have not been thoroughly explored by behavioral and computational research (with some notable exceptions; e.g. Hutchins, 1998; Bergen, 2004). Recently, Otis and Sagi (2008) used the semantic density of the cluster of words sharing a phonaestheme as a measure of

the strength of the relationship between the phonetic cluster and its proposed meaning.

Otis and Sagi used a corpus derived from Project Gutenberg (http://www.gutenberg.org/) as the basis for their analysis. Specifically, they used the bulk of the English language literary works available through the project's website. This resulted in a corpus of 4034 separate documents consisting of over 290 million words.

The bulk of the candidate phonaesthemes they tested were taken from the list used by Hutchins (1998), with the addition of two candidate phonaesthemes (*kn-* and *-ign*). Two letter combinations that were considered unlikely to be phonaesthemes (*br-* and *z-*) were also included in order to test the method's capacity for discriminating between phonaesthemes and non-phonaesthemes. Overall Otis and Sagi (2008) examined 47 possible phonaesthemes.

In cases where a phonetic cluster represents a phonaestheme, it intuitively follows that pairs of words sharing that phonetic cluster are more likely to share some aspect of their meaning than pairs of words chosen at random. Otis and Sagi tested whether this was true for any specific candidate phonaestheme using a Monte-Carlo analysis. First they identified all of the words in the corpus sharing a conjectured phonaestheme[7] and chose the most frequent representative word form for each stem, resulting in a cluster of word types representing each candidate phonaestheme.[8] Next they tested the statistical significance of this relationship by running 100 *t*-test comparisons. Each of these tests compared the relationship of 50 pairs of words chosen at random from the conjectured cluster with 50 pairs of words chosen at random from a similarly sized cluster, randomly generated from the entire corpus. The number of times these *t*-tests resulted in a statistically significant difference ($\alpha = .05$) was recorded. This analysis was repeated 3 times for each conjectured phonaestheme and the median value was used as the final result.

To determine whether a conjectured phonaestheme was statistically supported by their analysis Otis and Sagi compared the overall frequency of statistically significant t-tests with the binomial distribution for their α (.05). After applying a Bonferroni correction for performing 50 comparisons, the threshold for statistical significance of the binomial test was for 14 t-tests out of 100 to turn out as significant, with a frequency of 13 being marginally significant. Therefore, if the significance frequency (*#Sig* below) of a candidate phonaestheme was 15 or higher, that phonaestheme was judged as being supported by statistical evidence. Significance frequencies of 13 and 14 were considered as indicative of a phonaestheme for which there was only marginal statistical support.

Among Hutchins' original list of 44 possible phonaesthemes, 26 were found to be statistically reliable and 2 were marginally reliable. Overall the results were in line with the empirical data collected by Hutchins. By way of comparing the two datasets, *#Sig* and Hutchins' average rating measure were well correlated ($r = .53$). Neither of the unlikely phonaestheme candidates we examined were statistically supported phonaesthemes ($\#Sig_{\text{br-}} = 6$; $\#Sig_{\text{z-}} = 5$), whereas both of our newly hypothesized phonaesthemes were statistically supported ($\#Sig_{\text{kn-}} = 28$; $\#Sig_{\text{-ign}} = 23$). In addition to being able to use this measure as a decision criterion as to whether a specific phonetic cluster might be phonaesthemic, it can also be used to compare the relative strength of two such clusters. For instance, in the Gutenberg corpus the phonaesthemic ending *–owl* (e.g., 'growl', 'howl'; *#Sig*=97) was comprised of a cluster of words that were more similar to one another than *–oop* (e.g., 'hoop', 'loop'; *#Sig*=32).

Such results can then be used to test the cognitive effects of phonaesthemes. For instance, following the comparison above, we might hypothesize that the word 'growl' might be a better semantic prime for 'howl' than the word 'hoop' is for the word 'loop'. In contrast, because a word-initial *br-* is not phonaesthemic, the word 'breeze' is unlikely to be a semantic prime for the word 'brick'. In addition, it might be interesting to combine the diachronic analysis from the previous section with the synchronic analysis in this section to investigate questions such as when and how phonaesthemes come to be part of a language and what factors might affect the strength of a phonaestheme.

4 Discussion

While the method presented in this paper is aimed towards quantifying semantic relation-

[7] It is important to note that due to the nature of a written corpus, the match was orthographical rather than phonetic. However, in most cases the two are highly congruent.

[8] Because, in this case, Otis and Sagi were not interested in temporal changes in meaning, they used the overall word vectors rather than look at each context individually. As a result, each of the vectors used in the analysis is based on occurrences in many different documents and there was no need to control for the variability of the documents.

ships that were previously difficult to quantify, it also raises an interesting theoretical issue, namely the relationship between the statistically computed semantic space and the actual semantic content of words. On the one hand, simulations based on Latent Semantic Analysis have been shown to correlate with cognitive factors such as the acquisition of vocabulary and the categorization of texts (cf. Landauer & Dumais, 1997). On the other hand, in reality speakers' use of language relies on more than simple patterns of word co-occurrence – For instance, we use syntactic structures and pragmatic reasoning to supplement the meaning of the individual lexemes we come across (e.g., Fodor, 1995; Grice, 1989 [1975]). It is therefore likely that while LSA captures some of the variability in meaning exhibited by words in context, it does not capture all of it. Indeed, there is a growing body of methods that propose to integrate these two disparate sources of linguistic information (e.g., Pado and Lapata, 2007; Widdows, 2003)

Certainly, the results reported in this paper suggest that enough of the meaning of words and contexts is captured to allow interesting inferences about semantic change and the relatedness of words to be drawn with a reasonable degree of certainty. However, it is possible that some important aspects of meaning are systematically ignored by the analysis. For instance, it remains to be seen whether this method can distinguish between processes like pejoration and amerlioration as they require a fine grained distinction between 'good' and 'bad' meanings.

Regardless of any such limitations, it is clear that important information about meaning can be gathered through a systematic analysis of the contexts in which words appear. Furthermore, phenomena such as the existence of phonaesthemes and the success of LSA in predicting vocabulary acquisition rates, suggest that the acquisition of new vocabulary involves the gleaning of the meaning of words through their context. The role of context in semantic change is therefore likely to be an active one – when a listener encounters a word they are unfamiliar with they are likely to use the context in which it appears, as well as its phonetic composition, as clues to its meaning. Furthermore, if a word is likewise encountered in context in which it is unlikely, this unexpected observation may induce the listener to adjust their representation of both the context and the word in order to increase the overall coherence of the utterance or sentence. As a result, it is possible that examining the contexts in which a word is used in different documents and time periods might be useful not only as a tool for examining the history of a semantic change but also as an instrument for predicting its future progress. Overall, this suggests a dynamic view of the field of semantics – semantics as an ever-changing landscape of meaning. In such a view, semantic change is the norm as the perceived meaning of words keeps shifting to accommodate the contexts in which they are used.

References

Bergen, B. (2004). The Psychological Reality of Phonaesthemes. *Language*, 80(2), 291-311.

Berry, M. W. (1992) *SVDPACK: A Fortran-77 software library for the sparse singular value decomposition.* Tech. Rep. CS-92-159, Knoxville, TN: University of Tennessee.

Berry, M. W., Do, T., O'Brien, G. Vijay, K. Varadh an, S. (1993) *SVDPACKC (Version 1.0) User's Guide*, Tech. Rep. UT-CS-93-194, Knoxville, TN: University of Tennessee.

Bloomfield, L. (1933). *Language*. New York, NY: Holt, Rinehart and Winston.

Campbell, L. (2004) *Historical linguistics: An introduction* 2nd ed. Cambridge, MA: The MIT Press.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990) Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41, 391-407.

Ellegård, A. (1953) The Auxiliary Do: the Establishment and Regulation of its Use in English. *Gothenburg Studies in English*, 2. Stockholm: Almqvist and Wiksell.

Firth, J. (1930) *Speech*. London: Oxford University Press.

Firth, J. (1957) *Papers in Linguistics, 1934-1951*, Oxford University Press.

Fodor, J. D. (1995) Comprehending sentence structure. In L. R. Gleitman and M. Liberman, (Eds.), *Invitation to Cognitive Science*, volume 1. MIT Press, Cambridge, MA. 209-246.

Forston, B. W. (2003) An Approach to Semantic Change. In B. D. Joseph and R. D. Janda (Eds.), *The Handbook of Historical Linguistics*. Malden, MA: Blackwell Publishing. 648-666.

Grice, H. P. (1989) [1975]. Logic and Conversation. In *Studies in the Way of Words*. Cambridge, MA: Harvard University Press. 22-40.

Halliday, M. A. K., & Hasan, R. (1976) *Cohesion in English*. London: Longman.

Hock, H. H., and Joseph, B. D. (1996) *Language History, Language Change, and Language Relationship: An Introduction to Historical and Comparative Linguistics*. Berlin: Mouton de Gruyter.

Hoey, M. (1991) *Patterns of Lexis in Text*. London: Oxford University Press.

Hutchins, S. S. (1998). The psychological reality, variability, and compositionality of English phonesthemes. *Dissertation Abstracts International*, 59(08), 4500B. (University Microfilms No. AAT 9901857).

Infomap [Computer Software]. (2007). http://infomap-nlp.sourceforge.net/ Stanford, CA.

Kaufmann, S. (2000) Second-order cohesion. *Computational Intelligence.* 16, 511-524.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104, 211-240.

Levin, E., Sharifi, M., & Ball, J. (2006) Evaluation of utility of LSA for word sense discrimination. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, New York City. 77-80.

Marcu, D (2003) Automatic Abstracting, *Encyclopedia of Library and Information Science*, Drake, M. A., ed. 245-256.

Otis K., & Sagi E. (2008) Phonaesthemes: A Corpora-based Analysis. In B. C. Love, K. McRae, & V. M. Sloutsky (Eds.), *Proceedings of the 30th Annual Conference of the Cognitive Science Society*. Austin, TX: Cognitive Science Society.

Pado, S. & Lapata, M. (2007) Dependency-based Construction of Semantic Space Models. *Computational Linguistics*, 33, 161-199.

Riedel E., Dexter S. L., Scharber C., Doering A. (2006) Experimental Evidence on the Effectiveness of Automated Essay Scoring in Teacher Education Cases. *Journal of Educational Computing Research*, 35, 267-287.

Rissanen, M. (1994) The Helsinki Corpus of English Texts. In Kytö, M., Rissanen, M. and Wright S. (eds), *Corpora Across the Centuries: Proceedings of the First International Colloquium on English Diachronic Corpora*. Amsterdam: Rodopi.

Schütze, H. (1996) *Ambiguity in language learning: computational and cognitive models*. CA: Stanford.

Schütze, H. (1998) Automatic word sense discrimination. *Computational Linguistics* 24(1):97-124.

Takayama, Y., Flournoy R., & Kaufmann, S. (1998) *Information Mapping: Concept-Based Information Retrieval Based on Word Associations*. CSLI Tech Report. CA: Stanford.

Takayama, Y., Flournoy, R., Kaufmann, S. & Peters, S. (1999). Information retrieval based on domain-specific word associations. In Cercone, N. and Naruedomkul K. (eds.), *Proceedings of the Pacific Association for Computational Linguistics (PACLING'99)*, Waterloo, Canada. 155-161.

Traugott, E. C. (1999) The Role of Pragmatics in Semantic Change. In J. Verschueren (ed.), *Pragmatics in 1998: Selected Papers from the 6th International Pragmatics Conference, vol. II*. Antwerp: International Pragmatics Association. 93-102.

Traugott, E. C. (2005) Semantic Change. In *Encyclopedia of Language and Linguistics*, 2nd ed., Brown K. ed. Oxford: Elsevier.

Traugott, E. C., and Dasher R. B. (2002) *Regularity in Semantic Change*. Cambridge: Cambridge University Press.

Widdows, D. (2003) Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *Proceedings of the joint Human Language Technology Conference and Annual Meeting of the North American Chapter of the Association for Computational Linguistics*. Edmonton, Canada: Wiemer-Hastings. 197-204.

Widdows, D. (2004) *Geometry and Meaning*. CSLI Publications, CA: Stanford.

Context-theoretic Semantics for Natural Language: an Overview

Daoud Clarke
University of Sussex
Falmer, Brighton, UK
daoud.clarke@gmail.com

Abstract

We present the context-theoretic framework, which provides a set of rules for the nature of composition of meaning based on the philosophy of *meaning as context*. Principally, in the framework the composition of the meaning of words can be represented as multiplication of their representative vectors, where multiplication is distributive with respect to the vector space.

We discuss the applicability of the framework to a range of techniques in natural language processing, including subsequence matching, the lexical entailment model of Dagan et al. (2005), vector-based representations of taxonomies, statistical parsing and the representation of uncertainty in logical semantics.

1 Introduction

Techniques such as latent semantic analysis (Deerwester et al., 1990) and its variants have been very successful in representing the meanings of words as vectors, yet there is currently no theory of natural language semantics that explains how we should compose these representations: what should the representation of a phrase be, given the representation of the words in the phrase? In this paper we present such a theory, which is based on the philosophy of *meaning as context*, as epitomised by the famous sayings of Wittgenstein (1953), "Meaning just *is* use" and Firth (1957), "You shall know a word by the company it keeps". For the sake of brevity we shall present only a summary of our research, which is described in full in (Clarke, 2007), and we give a simplified version of the framework, which nevertheless suffices for the examples which follow.

We believe that the development of theories that can take vector representations of meaning beyond the word level, to the phrasal and sentence levels and beyond are essential for vector based semantics to truly compete with logical semantics, both in their academic standing and in application to real problems in natural language processing. Moreover the time is ripe for such a theory: never has there been such an abundance of immediately available textual data (in the form of the world-wide web) or cheap computing power to enable vector-based representations of meaning to be obtained. The need to organise and understand the new abundance of data makes these techniques all the more attractive since meanings are determined automatically and are thus more robust in comparison to hand-built representations of meaning. A guiding theory of vector based semantics would undoubtedly be invaluable in the application of these representations to problems in natural language processing.

The context-theoretic framework does not provide a formula for how to compose meaning; rather it provides mathematical guidelines for theories of meaning. It describes the nature of the vector space in which meanings live, gives some restrictions on how meanings compose, and provides us with a measure of the degree of entailment between strings for any implementation of the framework.

The remainder of the paper is structured as follows: in Section 2 we present the framework; in Section 3 we present applications of the framework:

- We describe subsequence matching (Section 3.1) and the lexical entailment model of (Dagan et al., 2005) (Section 3.2), both of which have been applied to the task of recognising textual entailment.
- We show how a vector based representation of a taxonomy incorporating probabilistic information about word meanings can be con-

Proceedings of the EACL 2009 Workshop on GEMS: GEometical Models of Natural Language Semantics, pages 112–119,
Athens, Greece, 31 March 2009.

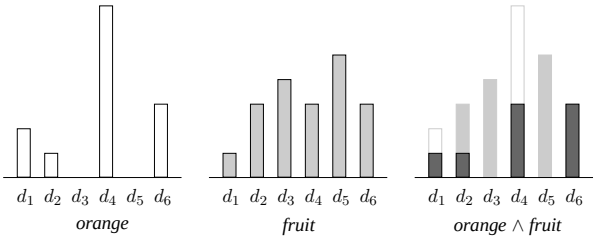


Figure 1: Vector representations of two terms in a space $L^1(S)$ where $S = \{d_1, d_2, d_3, d_4, d_5, d_6\}$ and their vector lattice meet (the darker shaded area).

structed in Section 3.3.

- We show how syntax can be represented within the framework in Section 3.4.
- We summarise our approach to representing uncertainty in logical semantics in Section 3.5.

2 Context-theoretic Framework

The context-theoretic framework is based on the idea that the vector representation of the meaning of a word is derived from the contexts in which it occurs. However it extends this idea to strings of any length: we assume there is some set S containing all the possible contexts associated with any string. A *context theory* is an implementation of the context-theoretic framework; a key requirement for a context theory is a mapping from strings to vectors formed from the set of contexts.

In vector based techniques, the set of contexts may be the set of possible dependency relations between words, or the set of documents in which strings may occur; in context-theoretic semantics however, the set of "contexts" can be any set. We continue to refer to it as a set of contexts since the intuition and philosophy which forms the basis for the framework derives from this idea; in practice the set may even consist of logical sentences describing the meanings of strings in model-theoretic terms.

An important aspect of vector-based techniques is measuring the frequency of occurrence of strings in each context. We model this in a general way as follows: let A be a set consisting of the words of the language under consideration. The first requirement of a context theory is a mapping $x \mapsto \hat{x}$ from a string $x \in A^*$ to a vector $\hat{x} \in L^1(S)^+$, where $L^1(S)$ means the set of all functions from S to the real numbers $\mathbb{R}$ which are finite under the L^1 norm,

$$\|u\|_1 = \sum_{s \in S} |u(s)|$$

and $L^1(S)^+$ restricts this to functions to the non-negative real numbers, $\mathbb{R}^+$; these functions are called the positive elements of the vector space $L^1(S)$. The requirement that the L^1 norm is finite, and that the map is only to positive elements reflects the fact that the vectors are intended to represent an estimate of relative frequency distributions of the strings over the contexts, since a frequency distribution will always satisfy these requirements. Note also that the l_1 norm of the context vector of a string is simply the sum of all its components and is thus proportional to its probability.

The set of functions $L^1(S)$ is a vector space under the point-wise operations:

$$\begin{aligned}(\alpha u)(s) &= \alpha u(s)\\ (u+v)(s) &= u(s) + v(s)\end{aligned}$$

for $u, v \in L^1(S)$ and $\alpha \in \mathbb{R}$, but it is also a lattice under the operations

$$\begin{aligned}(u \wedge v)(s) &= \min(u(s), v(s))\\ (u \vee v)(s) &= \max(u(s), v(s)).\end{aligned}$$

In fact it is a *vector lattice* or *Riesz space* (Aliprantis and Burkinshaw, 1985) since it satisfies the following relationships

$$\begin{aligned}&\text{if } u \leq v \text{ then } \alpha u \leq \alpha v\\ &\text{if } u \leq v \text{ then } u + w \leq v + w,\end{aligned}$$

where $\alpha \in \mathbb{R}^+$ and $\leq$ is the partial ordering associated with the lattice operations, defined by $u \leq v$ if $u \wedge v = u$.

Together with the l_1 norm, the vector lattice defines an *Abstract Lebesgue space* (Abramovich and Aliprantis, 2002) a vector space incorporating all the properties of a measure space, and thus can also be thought of as defining a probability space, where $\vee$ and $\wedge$ correspond to the union and intersection of events in the σ algebra, and the norm corresponds to the (un-normalised) probability.

2.1 Distributional Generality

The vector lattice nature of the space under consideration is important in the context-theoretic framework since it is used to define a degree of entailment between strings. Our notion of entailment is

based on the concept of *distributional generality* (Weeds et al., 2004), a generalisation of the distributional hypothesis of Harris (1985), in which it is assumed that terms with a more general meaning will occur in a wider array of contexts, an idea later developed by Geffet and Dagan (2005). Weeds et al. (2004) also found that frequency played a large role in determining the direction of entailment, with the more general term often occurring more frequently. The partial ordering of the vector lattice encapsulates these properties since $\hat{x} \leq \hat{y}$ if and only if y occurs more frequently in all the contexts in which x occurs.

This partial ordering is a strict relationship, however, that is unlikely to exist between any two given vectors. Because of this, we define a *degree of entailment*

$$\mathrm{Ent}(u, v) = \frac{\|u \wedge v\|_1}{\|u\|_1}.$$

This value has the properties of a conditional probability; in the case of $u = \hat{x}$ and $v = \hat{y}$ it is a measure of the degree to which the contexts string x occurs in are shared by the contexts string y occurs in.

2.2 Multiplication

The map from strings to vectors already tells us everything we need to know about the composition of words: given two words x and y, we have their individual context vectors $\hat{x}$ and $\hat{y}$, and the meaning of the string xy is represented by the vector $\widehat{xy}$. The question we address is what relationship should be imposed between the representation of the meanings of individual words $\hat{x}$ and $\hat{y}$ and the meaning of their composition $\widehat{xy}$. As it stands, we have little guidance on what maps from strings to context vectors are appropriate.

The first restriction we propose is that vector representations of meanings should be composable *in their own right*, without consideration of what words they originated from. In fact we place a strong requirement on the nature of multiplication on elements: we require that the multiplication $\cdot$ on the vector space defines a *lattice-ordered algebra*. This means that multiplication is associative, distributive with respect to addition, and satisfies $u \cdot v \geq 0$ if $u \geq 0$ and $v \geq 0$, i.e. the product of positive elements is also positive.

We argue that composition of context vectors needs to be compatible with concatenation of words, i.e.

$$\hat{x} \cdot \hat{y} = \widehat{xy},$$

i.e. the map from strings to context vectors defines a semigroup homomorphism. Then the requirement that multiplication is associative can be seen to be a natural one since the homomorphism enforces this requirement for context vectors. Similarly since all context vectors are positive their product in the algebra must also be positive, thus it is natural to extend this to all elements of the algebra. The requirement for distributivity is justified by our own model of meaning as context in text corpora, described in full elsewhere.

2.3 Context Theory

The above requirements give us all we need to define a context theory.

Definition 1 (Context theory)**.** *$\langle A, S, \hat{}, \cdot \rangle$ defines a context theory if $L^1(S)$ is a lattice-ordered algebra under the multiplication defined by $\cdot$ and $\hat{}$ defines a semigroup homomorphism $x \mapsto \hat{x}$ from A^* to $L^1(S)^+$.*

3 Context Theories for Natural Language

In this section we describe applications of the context-theoretic framework to applications in computational linguistics and natural language processing. We shall commonly use a construction in which there is a binary operation $\circ$ on S that makes it a semigroup. In this case $L^1(S)$ is a lattice-ordered algebra with convolution as multiplication:

$$(u \cdot v)(r) = \sum_{s \circ t = r} u(s)v(t)$$

for $r, s, t \in S$ and $u, v \in L^1(S)$. We denote the unit basis element associated with an element $x \in S$ by e_x, that is $e_x(y) = 1$ if and only if $y = x$, otherwise $e_x(y) = 0$.

3.1 Subsequence Matching

A string $x \in A^*$ is called a "subsequence" of $y \in A^*$ if each element of x occurs in y in the same order, but with the possibility of other elements occurring in between, so for example $abba$ is a subsequence of $acabcba$ in $\{a, b, c\}^*$. We denote the set of subsequences of x (including the empty string) by $\mathrm{Sub}(x)$. Subsequence matching compares the subsequences of two strings: the

more subsequences they have in common the more similar they are assumed to be. This idea has been used successfully in text classification (Lodhi et al., 2002) and recognising textual entailment (Clarke, 2006).

We can describe such models using a context theory $\langle A, A^*, \hat{}, \cdot \rangle$, where $\cdot$ is convolution in $L^1(A^*)$ and

$$\hat{x} = (1/2^{|x|}) \sum_{y \in \mathrm{Sub}(x)} e_y,$$

i.e. the context vector of a string is a weighted sum of its subsequences. Under this context theory $\hat{x} \leq \hat{y}$, i.e. x completely entails y if x is a subsequence of y.

Many variations on this context theory are possible, for example using more complex mappings to $L^1(A^*)$. The context theory can also be adapted to incorporate a measure of lexical overlap between strings, an approach that, although simple, performs comparably to more complex techniques in tasks such as recognising textual entailment (Dagan et al., 2005)

3.2 Lexical Entailment Model

Glickman and Dagan (2005) define their own model of entailment and apply it to the task of recognising textual entailment. They estimate entailment between words based on occurrences in documents: they estimate a *lexical entailment probability* $\textsc{lep}(x, y)$ between two terms x and y to be

$$\textsc{lep}(x, y) \simeq \frac{n_{x,y}}{n_y}$$

where n_y and $n_{x,y}$ denote the number of documents that the word y occurs in and the words x and y both occur in respectively.

We can describe this using a context theory $\langle A, D, \hat{}, \cdot \rangle$, where D is the set of documents, and

$$\hat{x}(d) = \begin{cases} 1 & \text{if } x \text{ occurs in document } d \\ 0 & \text{otherwise.} \end{cases}.$$

In this case the estimate of $\textsc{lep}(x, y)$ coincides with our own degree of entailment $\mathrm{Ent}(x, y)$.

There are many ways in which the multiplication $\cdot$ can be defined on $L^1(D)$. The simplest one defines $e_d \cdot e_f = e_d$ if $d = f$ and $e_d e_f = 0$ otherwise. The effect of multiplication of the context vectors of two strings is then set intersection:

$$(\hat{x} \cdot \hat{y})(d) = \begin{cases} 1 & \text{if } x \text{ and } y \text{ occur in document } d \\ 0 & \text{otherwise.} \end{cases}$$

Model	Accuracy	CWS
Dirichlet (10^6)	0.584	0.630
Dirichlet (10^7)	0.576	0.642
Bayer (MITRE)	0.586	0.617
Glickman (Bar Ilan)	0.586	0.572
Jijkoun (Amsterdam)	0.552	0.559
Newman (Dublin)	0.565	0.6

Table 1: Results obtained with our Latent Dirichlet projection model on the data from the first Recognising Textual Entailment Challenge for two document lengths $N = 10^6$ and $N = 10^7$ using a cut-off for the degree of entailment of 0.5 at which entailment was regarded as holding. CWS is the confidence weighted score — see (Dagan et al., 2005) for the definition.

Glickman and Dagan (2005) do not use this measure, possibly because the problem of data sparseness makes it useless for long strings. However the measure they use can be viewed as an approximation to this context theory.

We have also used this idea to determine entailment, using latent Dirichlet allocation to get around the problem of data sparseness. A model was built using a subset of around 380,000 documents from the Gigaword corpus, and the model was evaluated on the dataset from the first Recognising Textual Entailment Challenge; the results are shown in Table 1. In order to use the model, a document length had to be chosen; it was found that very long documents yielded better performance at this task.

3.3 Representing Taxonomies

In this section we describe how the relationships described by a taxonomy, the collection of **is-a** relationships described by ontologies such as WordNet (Fellbaum, 1989), can be embedded in the vector lattice structure that is crucial to the context-theoretic framework. This opens up the way to the possibility of new techniques that combine the vector-based representations of word meanings with the ontological ones, for example:

- **Semantic smoothing** could be applied to vector based representations of an ontology, for example using distributional similarity measures to move words that are distributionally similar closer to each other in the vector space. This type of technique may allow the

benefits of vector based techniques and ontologies to be combined.

- **Automatic classification:** representing the taxonomy in a vector space may make it easier to look for relationships between the meanings in the taxonomy and meanings derived from vector based techniques such as latent semantic analysis, potentially aiding in classifying word meanings in a taxonomy.

- The new vector representation could lead to new measures of **semantic distance**, for example, the L^p norms can all be used to measure distance between the vector representations of meanings in a taxonomy. Moreover, the vector-based representation allows ambiguity to be represented by adding the weighted representations of individual senses.

We assume that the **is-a** relation is a partial ordering; this is true for many ontologies. We wish to incorporate the partial ordering of the taxonomy into the partial ordering of the vector lattice. We will make use of the following result relating to partial orders:

Definition 2 (Ideals)**.** *A* lower set *in a partially ordered set S is a set T such that for all $x, y \in S$, if $x \in T$ and $y \leq x$ then $y \in T$.*

The principal ideal generated by an element x *in a partially ordered set S is defined to be the lower set $\downarrow(x) = \{y \in S : y \leq x\}$.*

Proposition 3 (Ideal Completion)**.** *If S is a partially ordered set, then $\downarrow(\cdot)$ can be considered as a function from S to the powerset 2^S. Under the partial ordering defined by set inclusion, the set of lower sets form a complete lattice, and $\downarrow(\cdot)$ is a completion of S, the* ideal completion.

We are also concerned with the probability of concepts. This is an idea that has come about through the introduction of "distance measures" on taxonomies (Resnik, 1995). Since terms can be ascribed probabilities based on their frequencies of occurrence in corpora, the concepts they refer to can similarly be assigned probabilities. The probability of a concept is the probability of encountering an instance of that concept in the corpus, that is, the probability that a term selected at random from the corpus has a meaning that is subsumed by that particular concept. This ensures that more general concepts are given higher probabilities, for example if there is a most general concept (a top-most node in the taxonomy, which may correspond for example to "entity") its probability will be one, since every term can be considered an instance of that concept.

We give a general definition based on this idea which does not require probabilities to be assigned based on corpus counts:

Definition 4 (Real Valued Taxonomy)**.** *A real valued taxonomy is a finite set S of* concepts *with a partial ordering $\leq$ and a positive real function p over S. The* measure *of a concept is then defined in terms of p as*

$$\hat{p}(x) = \sum_{y \in \downarrow(x)} p(y).$$

The taxonomy is called probabilistic *if $\sum_{x \in S} p(s) = 1$. In this case $\hat{p}$ refers to the* probability of a concept.

Thus in a probabilistic taxonomy, the function p corresponds to the probability that a term is observed whose meaning corresponds (in that context) to that concept. The function $\hat{p}$ denotes the probability that a term is observed whose meaning in that context is subsumed by the concept.

Note that if S has a top element I then in the probabilistic case, clearly $\hat{p}(I) = 1$. In studies of distance measures on ontologies, the concepts in S often correspond to senses of terms, in this case the function p represents the (normalised) probability that a given term will occur with the sense indicated by the concept. The top-most concept often exists, and may be something with the meaning "entity"—intended to include the meaning of all concepts below it.

The most simple completion we consider is into the vector lattice $L^1(S)$, with basis elements $\{e_x : x \in S\}$.

Proposition 5 (Ideal Vector Completion)**.** *Let S be a probabilistic taxonomy with probability distribution function p that is non-zero everywhere on S. The function ψ from S to $L^1(S)$ defined by*

$$\psi(x) = \sum_{y \in \downarrow(x)} p(y) e_y$$

is a completion of the partial ordering of S under the vector lattice order of $L^1(S)$, satisfying $\|\psi(x)\|_1 = \hat{p}(x)$.

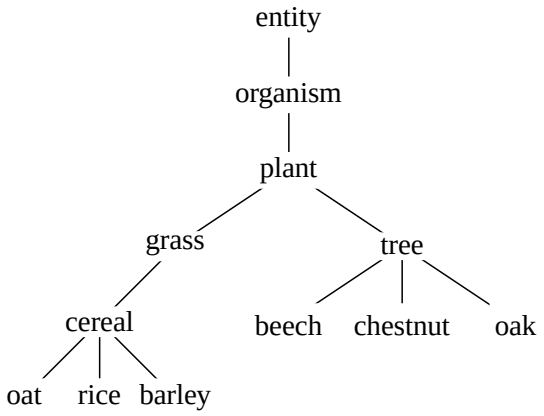


Figure 2: A small example taxonomy extracted from WordNet (Fellbaum, 1989).

Proof. The function ψ is clearly order-preserving: if $x \leq y$ in S then since $\downarrow(x) \subseteq \downarrow(y)$, necessarily $\psi(x) \leq \psi(y)$. Conversely, the only way that $\psi(x) \leq \psi(y)$ can be true is if $\downarrow(x) \subseteq \downarrow(y)$ since p is non-zero everywhere. If this is the case, then $x \leq y$ by the nature of the ideal completion. Thus ψ is an order-embedding, and since $L^1(S)$ is a complete lattice, it is also a completion. Finally, note that $\|\psi(x)\|_1 = \sum_{y \in \downarrow(x)} p(y) = \hat{p}(x)$. □

This completion allows us to represent concepts as elements within a vector lattice so that not only the partial ordering of the taxonomy is preserved, but the probability of concepts is also preserved as the size of the vector under the L^1 norm.

3.4 Representing Syntax

In this section we give a description link grammar (Sleator and Temperley, 1991) in terms of a context theory. Link grammar is a lexicalised syntactic formalism which describes properties of words in terms of *links* formed between them, and which is context-free in terms of its generative power; for the sake of brevity we omit the details, although a sample link grammar parse is show in Figure 3.

Our formulation of link grammar as a context theory makes use of a construction called a *free inverse semigroup*. Informally, the free inverse semigroup on a set S is formed from elements of S and their inverses, $S^{-1} = \{s^{-1} : s \in S\}$, satisfying no other condition than those of an inverse semigroup. Formally, the free inverse semigroup is defined in terms of a congruence relation on $(S \cup S^{-1})^*$ specifying the inverse property and commutativity of idempotents — see (Munn, 1974) for details. We denote the free inverse semigroup on S by $\mathrm{FIS}(S)$.

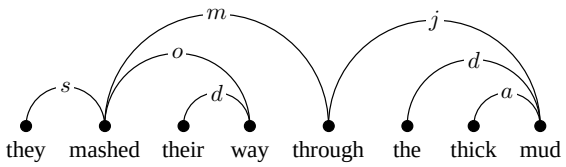


Figure 3: A link grammar parse. Link types: s: subject, o: object, m: modifying phrases, a: adjective, j: preposition, d: determiner.

Free inverse semigroups were shown by Munn (1974) to be equivalent to *birooted word trees*. A birooted word-tree on a set A is a directed acyclic graph whose edges are labelled by elements of A which does not contain any subgraphs of the form $\bullet \xrightarrow{a} \bullet \xleftarrow{a} \bullet$ or $\bullet \xleftarrow{a} \bullet \xrightarrow{a} \bullet$, together with two distinguished nodes, called the start node, $\square$ and finish node, $\circ$.

An element in the free semigroup $\mathrm{FIS}(S)$ is denoted as a sequence $x_1^{d_1} x_2^{d_2} \ldots x_n^{d_n}$ where $x_i \in S$ and $d_i \in \{1, -1\}$.

We construct the birooted word tree by starting with a single node as the start node, and for each i from 1 to n:

- Determine if there is an edge labelled x_i leaving the current node if $d_i = 1$, or arriving at the current node if $d_i = -1$.
- If so, follow this edge and make the resulting node the current node.
- If not, create a new node and join it with an edge labelled x_i in the appropriate direction, and make this node the current node.

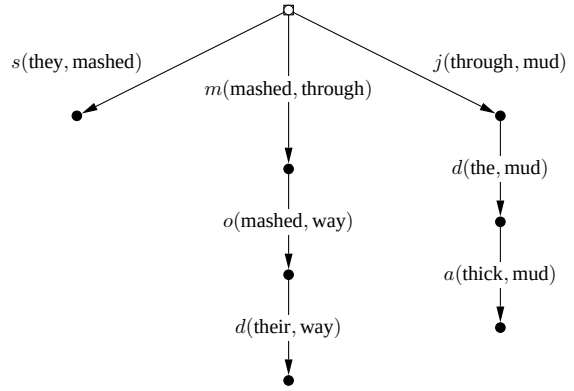
The finish node is the current node after the n iterations.

The product of two elements x and y in the free inverse semigroup can be computed by finding the birooted word-tree of x and that of y, joining the graphs by equating the start node of y with the finish node of x (and making it a normal node), and merging any other nodes and edges necessary to remove any subgraphs of the form $\bullet \xrightarrow{a} \bullet \xleftarrow{a} \bullet$ or $\bullet \xleftarrow{a} \bullet \xrightarrow{a} \bullet$. The inverse of an element has the same graph with start and finish nodes exchanged.

We can represent parses of sentences in link grammar by translating words to syntactic categories in the *free inverse semigroup*. The parse shown earlier for "they mashed their way through the thick mud" can be represented in the inverse semigroup on $S = \{s, m, o, d, j, a\}$ as

$$ss^{-1}modd^{-1}o^{-1}m^{-1}jdaa^{-1}d^{-1}j^{-1}$$

which has the following birooted word-tree (the words which the links derive from are shown in brackets):



Let A be the set of words in the natural language under consideration, S be the set of link types. Then we can form a context theory $\langle A, \mathrm{FIS}(S), \hat{}, \cdot \rangle$ where $\cdot$ is multiplication defined by convolution on $\mathrm{FIS}(S)$, and a word $a \in A$ is mapped to a probabilistic sum $\hat{a}$ of its link possible grammar representations (called *disjuncts*). Thus we have a context theory which maps a string x to elements of $L^1(\mathrm{FIS}(S))$; if there is a parse for this string then there will be some component of $\hat{x}$ which corresponds to an idempotent element of $\mathrm{FIS}(S)$. Moreover we can interpret the magnitude of the component as the probability of that particular parse, thus the context theory describes a probabilistic variation of link grammar.

3.5 Uncertainty in Logical Semantics

For the sake of brevity, we summarise our approach to representing uncertainty in logical semantics, which is described in full elsewhere. Our aim is to be able to reason with probabilistic information about uncertainty in logical semantics. For example, in order to represent a natural language sentence as a logical statement, it is necessary to parse it, which may well be with a statistical parser. We may have hundreds of possible parses and logical representations of a sentence, and associated probabilities. Alternatively, we may wish to describe our uncertainty about word-sense disambiguation in the representation. Incorporating such probabilistic information into the representation of meaning may lead to more robust systems which are able to cope when one component fails.

The basic principle we propose is to first represent unambiguous logical statements as a context theory. Our uncertainty about the meaning of a sentence can then be represented as a probability distribution over logical statements, whether the uncertainty arises from parsing, word-sense disambiguation or any other source. Incorporating this information is then straightforward: the representation of the sentence is the weighted sum of the representation of each possible meaning, where the weights are given by the probability distribution.

Computing the degree of entailment using this approach is computationally challenging, however we have shown that it is possible to estimate the degree of entailment by computing a lower bound on this value by calculating pairwise degrees of entailment for each possible logical statement.

4 Related Work

Mitchell and Lapata (2008) proposed a framework for composing meaning that is extremely general in nature: there is no requirement for linearity in the composition function, although in practice the authors do adopt this assumption. Indeed their "multiplicative models" require composition of two vectors to be a linear function of their tensor product; this is equivalent to our requirement of distributivity with respect to vector space addition.

Various ways of composing vector based representations of meaning were investigated by Widdows (2008), including the tensor product and direct sum. Both of these are compatible with the context theoretic framework since they are distributive with respect to the vector space addition.

Clark et al. (2008) proposed a method of composing meaning that generalises Montague semantics; further work is required to determine how their method of composition relates to the context-theoretic framework.

Erk and Pado (2008) describe a method of composition that allows the incorporation of selectional preferences; again further work is required to determine the relation between this work and the context-theoretic framework.

5 Conclusion

We have given an introduction to the context-theoretic framework, which provides mathematical guidelines on how vector-based representations of meaning should be composed, how entailment should be determined between these representations, and how probabilistic information should be incorporated.

We have shown how the framework can be applied to a wide range of problems in computational linguistics, including subsequence matching, vector based representations of taxonomies and statistical parsing. The ideas we have presented here are only a fraction of those described in full in (Clarke, 2007), and we believe that even that is only the tip of the iceberg with regards to what it is possible to achieve with the framework.

Acknowledgments

I am very grateful to my supervisor David Weir for all his help in the development of these ideas, and to Rudi Lutz and the anonymous reviewers for many useful comments and suggestions.

References

Y. A. Abramovich and Charalambos D. Aliprantis. 2002. *An Invitation to Operator Theory*. American Mathematical Society.

Charalambos D. Aliprantis and Owen Burkinshaw. 1985. *Positive Operators*. Academic Press.

Stephen Clark, Bob Coecke, and Mehrnoosh Sadrzadeh. 2008. A compositional distributional model of meaning. In *Proceedings of the Second Symposium on Quantum Interaction, Oxford, UK*, pages 133–140.

Daoud Clarke. 2006. Meaning as context and subsequence analysis for textual entailment. In *Proceedings of the Second PASCAL Recognising Textual Entailment Challenge*.

Daoud Clarke. 2007. *Context-theoretic Semantics for Natural Language: an Algebraic Framework*. Ph.D. thesis, Department of Informatics, University of Sussex.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

Scott Deerwester, Susan Dumais, George Furnas, Thomas Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Katrin Erk and Sebastian Pado. 2008. A structured vector space model for word meaning in context. In *Proceedings of EMNLP*.

Christaine Fellbaum, editor. 1989. *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, Massachusetts.

John R. Firth. 1957. Modes of meaning. In *Papers in Linguistics 1934–1951*. Oxford University Press, London.

Maayan Geffet and Ido Dagan. 2005. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, University of Michigan.

Oren Glickman and Ido Dagan. 2005. A probabilistic setting and lexical cooccurrence model for textual entailment. In *ACL-05 Workshop on Empirical Modeling of Semantic Equivalence and Entailment*.

Zellig Harris. 1985. Distributional structure. In Jerrold J. Katz, editor, *The Philosophy of Linguistics*, pages 26–47. Oxford University Press.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. 2002. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244, Columbus, Ohio, June. Association for Computational Linguistics.

W. D. Munn. 1974. Free inverse semigroup. *Proceedings of the London Mathematical Society*, 29:385–404.

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *IJCAI*, pages 448–453.

Daniel D. Sleator and Davy Temperley. 1991. Parsing english with a link grammar. Technical Report CMU-CS-91-196, Department of Computer Science, Carnegie Mellon University.

Julie Weeds, David Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. In *Proceedings of the 20th International Conference of Computational Linguistics, COLING-2004, Geneva, Switzerland.*

Dominic Widdows. 2008. Semantic vector products: Some initial investigations. In *Proceedings of the Second Symposium on Quantum Interaction, Oxford, UK*.

Ludwig Wittgenstein. 1953. *Philosophical Investigations*. Macmillan, New York. G. Anscombe, translator.

Author Index