The Dissertation Committee for Joseph Simon Reisinger
certifies that this is the approved version of the following dissertation:

Latent Variable Models of Distributional Lexical Semantics

Committee:

Raymond J. Mooney, Supervisor

Katrin Erk

Joydeep Ghosh

Fernando Pereira

Pradeep Ravikumar

Latent Variable Models of Distributional Lexical Semantics

by

Joseph Simon Reisinger, B.S. Math.; B.S. CS.; M.S. CS.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2012

For A

Acknowledgments

In roughly chronological order, large chunks of this thesis were conceived and drafted at: Metro, Spiderhouse, La Tazza Fresca, Little City, JPs Java, Epoch, Pause Cafe, Ritual, Clementine, Bennu, Mozart's, Thunderbird, Little City (downtown), Medici, Cherrywood, Bouldin Creek, Dominican Joe's, Coffee Bar, The Terminal, Matome Cafe, Bousingot, Stumptown, and Sightglass.

Folks at IBM T.J. Watson, particularly Mahesh Viswanathan, Roberto Sicconi and Liam Comerford; also Glenn Hui and Pedro DeRose. I suppose one has very few chances in life to work in an Eero Saarinen space-arc.

Google fundamentally shaped my research, both inside and outside academia. I owe a ton to my officemates Benjamin Van Durme, Sujith Ravi, Partha Talukdar and Mark Dredze for introducing me to NLP research. Sugato Basu and Roberto Bayardo let me play around with their super-secret data for months, resulting in a model that was only just recently published. Fernando Pereira for some very illuminating top-down discussions about the limits of information extraction. Finally I owe terrible debt to Marius Paşca for imparting his wisdom about reward travel and for putting up with various crazy project ideas while I interned with him at Google (3 times!).

A wonderful bunch of folks I introduced myself to after catching them gorging on wild berries near the Cam, all of whom have gone on to do amazing things:

Liz Bonawitz, Mike Frank, Steve Piantadosi and Tomer Ullman. The bulk of this thesis stems directly or indirectly from that encounter.

Various faculty at UT: Risto Miikkulainen, Benjamin Kuipers, Peter Stone, Jason Baldridge and Lorenzo Alvisi.[1]

Reuben Grinberg for his tough and devoted critiques of my NSF application, which is quite probably the main reason it ever got accepted. That in turn gave me enough confidence to see the grad school thing through.

Coffee-shop junkies and friends in Austin: Yinon Bentor, Gilbert Bernstein[2], Yonatan Bisk[3], Leif Johnson, Ann Kilzer, Scott Kirkpatrick[4], Brad Knox, Andrew Matsuoka and Edmund Wong.[5]

Team SAM: Austin Waters and Bryan Silverthorn. They know more about VEM, weighted averages on $\mathbb{S}^{D-1}$ and wrapped Gaussians than you.

My awesome thesis committee for helping put the finishing touches on this work: Katrin Erk, Joydeep Ghosh, Fernando Pereira, and Pradeep Ravkumar.

My advisor Ray Mooney, for even-handedly focusing all the wild ideas that I've come up with over the years into a coherent thesis.

Finally my family, particularly my brothers, Nathaniel and Jim and my parents Claudia and Joe. I love you guys.

[1]For patiently explaining How Things Work
[2]For being able to discuss tropical geometry and the ethnology of tramps in the same breath.
[3]For being himself.
[4]Whose name is now immortally misspelled in print.
[5]For many mutual grad school therapy sessions, debating the finer-points of PhD student life.

Latent Variable Models of Distributional Lexical Semantics

Publication No. _____

Joseph Simon Reisinger, Ph.D.
The University of Texas at Austin, 2012

Supervisor: Raymond J. Mooney

In order to respond to increasing demand for natural language interfaces—and provide meaningful insight into user query intent—fast, scalable lexical semantic models with flexible representations are needed. Human concept organization is a rich phenomenon that has yet to be accounted for by a single coherent psychological framework: Concept generalization is captured by a mixture of prototype and exemplar models, and local taxonomic information is available through multiple overlapping organizational systems. Previous work in computational linguistics on extracting lexical semantic information from unannotated corpora does not provide adequate representational flexibility and hence fails to capture the full extent of human conceptual knowledge. In this thesis I outline a family of probabilistic models capable of capturing important aspects of the rich organizational structure found in human language that can predict contextual variation, selectional preference and feature-saliency norms to a much higher degree of accuracy than

previous approaches. These models account for cross-cutting structure of concept organization—i.e. *selective attention*, or the notion that humans make use of different categorization systems for different kinds of generalization tasks—and can be applied to Web-scale corpora. Using these models, natural language systems will be able to infer a more comprehensive semantic relations, which in turn may yield improved systems for question answering, text classification, machine translation, and information retrieval.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis is concerned with the foundational Natural Language Processing task of *lexical semantics*, or "the study of how and what the words of a language denote" (Pustejovsky, 1995). Lexical semantics can be viewed as an interstitial field between the study of *syntax*, or the structure of language, and *semantics* the study of meaning. The question of what words mean is also closely related to the question of what concepts can have names, suggesting approaches motivated by cognitive science (Murphy, 2002).

One particularly compelling lexical semantic theory is the *distributional* hypothesis, which states that the observed pattern of word co-occurrence in text captures elements of word meaning. This hypothesis suggests a "theory of meaning" that can be operationalized into an empirical procedure for obtaining semantic representations directly from large textual corpora (Cruse, 1986). Given the availability of massive textual corpora e.g. from the Web, a significant amount of recent work in empirical Natural Language Processing (NLP) seeks to leverage this hypothesis (see e.g. (Turney and Pantel, 2010) for a recent overview).

Distributional lexical semantics arose at the confluence of distributional analysis in structuralist linguistics (Harris, 1954), British corpus linguistics (Firth,

1957) and psychology (Miller and Charles, 1991). In particular, Firth's "context of situation" theory succinctly captures the context-dependent nature of meaning:

> You shall know a word by the company it keeps

(Firth, 1957), cf. (Palmer, 1976). This notion of *collocation* captures predictable aspects (in the statistical sense) of word usage and hence may be potentially applied for insight into the underlying meaning.

Nida (1975) demonstrates the power of context with a thought experiment involving the artificial word *tezgüino*:

- A bottle of *tezgüino* is on the table.
- Everyone likes *tezgüino*.
- *Tezgüino* makes you drunk.
- We make *tezgüino* out of corn.

Taking this contextual evidence together, it can be inferred that *tezgüino* is an alcoholic beverage made out of corn mash (cf. Lin, 1998).

In humans, such contextual priming often takes place given only a single example occurrence of a new word. McDonald and Ramscar (2001) propose the example:

- He filled the *wampimuk* with the substance, passed it around and we all drunk some.

- We found a little, hairy *wampimuk* sleeping behind the tree.

In this case, the sentential context surrounding *wapimuk* is sufficient to capture fine-grained details about its meaning: either (1) a kind of drinking container or (2) a small mammal. Given the considerable amount of information contained within such word occurrence contexts, modern modern sources of textual data such as the Web pose a tremendous opportunity, since they contain easily hundreds of billions of such examples.

1.1 Distributional Lexical Semantics

Drawing features from local occurrence contexts (e.g. *wapimuk* might have a feature indicating that it is the object of the verb *filled*, or the subject of the verb *sleep*), word meaning can be represented as high-dimensional vectors inhabiting a common space, where each (sparse) feature dimension captures a semantic or syntactic property of interest (e.g. Erk and Pado, 2008; Lowe, 2001; Turney and Pantel, 2010). Such *distributional* representations of meaning can be used to induce measures of word similarity that are useful in a variety of tasks such as information retrieval (Manning et al., 2008), large-scale taxonomy induction (Snow et al., 2006), and knowledge acquisition (Van Durme and Paşca, 2008).

Although they lack proper formal logical semantics (cf. Montague, 1973), distributional methods have been adopted widely due to their relative simplicity and high scalability (Gorman and Curran, 2006; Curran, 2004a; Padó and Lapata, 2007; Pereira et al., 1993; Schütze, 1998a; Turney, 2006; Thater et al., 2010), trading off

representational precision and formal consistency for improved coverage and applicability. Indeed early approaches to distributional lexical semantics did not take into account compositionality (Landauer and Dumais, 1997; Lund and Burgess, 1996) or lexical ambiguity.

Typical approaches to distributional lexical semantics generate word representations that conflate multiple senses and usages. In recent work, there have been numerous approaches for addressing these shortcomings. For example, Schütze (1998b) and McCarthy and Carroll (2003a) use unsupervised clustering methods to perform word-sense disambiguation, and contextual dependence has been addressed using context word-priming (McDonald and Brew, 2004), direct vector composition (Mitchell and Lapata, 2008), exemplar activation (Erk and Padó, 2010), and syntactically-enriched vector spaces (Thater et al., 2010).

In this thesis I argue for a more general synthesis of psychological models of concept organization and computational linguistic models of distributional lexical semantics, focusing particularly on modeling context-dependence in word meaning. In particular, I draw parallels between the role of context-dependence in determining word meaning and *selective attention* in models of concept organization (Ross and Murphy, 1999). Selective attention refers to the observation that humans attend to different features of a stimuli in different contexts and categorize objects differently depending on which features are actively attended to.

Towards this end, I introduce a progression of probabilistic *latent variable models* whose underlying structure can be used to model selective attention, while preserving the simplicity and scalability of traditional distributional approaches.

1.2 Latent Variable Models of Lexical Semantics

The primary goal of this thesis is to introduce scalable lexical semantic models that can account for the fine-grained structure of word relatedness, capturing the multidimensional nature of relations between lexical units and their context-dependence. For example, computing the semantic similarity between *wine* and *vinegar* should only take into account a small number salient features of those concepts (in line with theories of selective attention), and those features should be quite different from those determining the similarity between *wine* and *bottle*, despite the fact that all three words occur in similar contexts.

Concepts and meanings in human language are organized in terms of complex assemblies of properties or features and exhibit rich structure (Ross and Murphy, 1999). Humans categorize objects using multiple orthogonal taxonomic structures, where generalization depends critically on what features are relevant to the particular structure (Smith and Shafto, 2011). For example, foods can be organized in terms of their nutritional value (high in fiber) or situationally (commonly eaten for Thanksgiving). Furthermore there is significant evidence for overlapping categorization systems in Wikipedia and WordNet (e.g. people are organized by occupation or by nationality). The effects of these overlapping categorization systems manifest themselves at the lexical semantic level (Murphy, 2002), implying that lexicographical word senses and traditional computational models of word-sense based on clustering or exemplar activation are too impoverished to capture the rich dynamics of word usage.

To model selective attention in a lexical semantic context, I introduce a uni-

fied set of probabilistic models based on *multi-view* clustering (e.g. Cross-cutting categorization Shafto et al., 2006), generalizing work on context-dependent distributional lexical semantics. Multiple clustering finds feature subsets (categorization systems) that produce high quality clusterings of the data. For example words might be clustered based on their syntactic (syntagmatic) or thematic (paradigmatic) usage, exposing multiple dimensions of word-relatedness. In particular, such models can be used to capture the *microstructure* of word relatedness, breaking up word features into multiple categorization systems and then computing similarity separately for each system. Furthermore, context-dependent variation in word usage can be accounted for naturally in these models by computing the joint likelihood of a term and its context given each cluster (Dinu and Lapata, 2010).

Capturing the multiple overlapping clustering structure of natural concepts could improve a range of Natural Language Processing (NLP) tasks: question answering (Tokunaga et al., 2005), unsupervised semantic parsing (Poon and Domingos, 2009), query intent classification and expansion (Jansen et al., 2007), coreference resolution (Haghighi and Klein, 2007) and textual entailment (Tatu and Moldovan, 2005).

1.3 Thesis Organization

Chapter 3 introduces the corpora and evaluation methodology employed in this thesis. In addition to various human-studies evaluations, four main empirical problems from lexical semantics are considered:

1. Modeling word association and typicality norms (**WS-353** §3.2.1 and **McRae** §3.2.3).

2. Modeling the selectional preference of verbs (**Selectional Preference** §3.2.2).

3. Modeling word relations in general (**BLESS**; 3.2.4).

4. Identifying suitable replacements for a target word in a given sentence (**Lexical Substitution**; §3.2.5).

Distributional models of lexical semantics typically create a single "prototype" vector to represent the meaning of a word. However, due to lexical ambiguity, encoding word meaning with a single vector is problematic. Chapter 4 introduces the family of **multi-prototype** models, which extend such models by first breaking word occurrences up over latent senses. Such models are capable of accounting for homonymy, as well as other forms of variation in word usage, like similar context-dependent methods (Erk and Pado, 2008). The set of vectors for a word is determined by unsupervised *word sense discovery* (Schütze, 1998a), which clusters the contexts in which a word appears. Average prototype vectors are then computed separately for each cluster, producing a distributed representation for each word. Multi-prototype models outperform standard single-prototype models on common lexical semantic tasks such as modeling word similarity and paraphrase prediction.

Although multi-prototype models can address context-dependence and ambiguity due to homonymy, they break down in cases where words are polysemous, or when capturing the *shared structure* between individual senses is important, e.g.

in tasks such as modeling the *selectional preference* of verbs (§4.5.3). To address this issue, I introduce a **tiered clustering** model (§4.1.2) that separates the features common to each word sense from the features that are most discriminative for that sense.

Moving beyond approaches based on multiple prototypes, I introduce a **Multi-view** model of lexical semantics capable fo capturing the full spectrum of relations between words and word-senses (Chapter 5). Such models can partition the space of word contexts across multiple clusterings, leading to more fine-grained models of contextual dependence, as well as vector-valued word relatedness. Sense proliferation itself may be in part due to the conflation of multiple organizational systems linking the target word to other similar words, leading to a large number of partially overlapping senses. By treating word sense in a multiple clustering framework, these organizational systems can be uncovered and leveraged to build models of per-word semantic generalization. Word senses can be organized along topical, syntactic or operational lines, and different organizational systems account for different subsets of the full set of lexicographical senses. Furthermore, each clustering defines a subset of the available features that are deemed salient, allowing models of lexical semantics the freedom to choose between several relevant subspaces and ignore irrelevant features.

Compared to simpler models, multi-view models of lexical semantics can account for context-dependent feature saliency and multiple dimensions of relatedness, leading to better performance on semantic similarity tasks.

1.4 Summary of Contributions

This thesis introduces and comprehensively evaluates three new classes of latent variable models for lexical semantics:

- **Multi-Prototype Model** – A model explicitly accounting for ambiguity in homonymous words based on clustering individual occurrences (Chapter 4).
- **Tiered Clustering Model** – An extension of the multi-prototype model capable of accounting for *shared* structure between word senses in polysemous words (Chapter 4).
- **Multi-View Model** – A model of *selective attention* in concepts capturing multiple feature subsets corresponding to different latent relational subspaces between words (Chapter 5).

Systematic evaluation of these models yields three main empirical results:

- Multi-prototype models outperforms single-prototype models on several word similarity tasks (§4.5).
- The tiered model outperforms the multi-prototype model for *selectional preference*, a task where modeling shared structured explicitly is necessary (§4.5.3).
- Several variants of the multi-view model outperform simpler baseline models on attribute, event, and hypernym recall as well as modeling lexical substitution for adverbs and adjectives (§5.3.3).

Chapter 2

Background and Related Work

This chapter summarizes relevant background work from distributional lexical semantics (§2.1), latent variable modeling (§2.2) and the psychology of concepts (§2.3).

2.1 Distributional Lexical Semantics

Word meaning can be represented as high-dimensional vectors inhabiting a common space whose feature dimensions capture semantic or syntactic properties of interest (e.g. Harper, 1965; Padó and Lapata, 2007; Schütze, 1998a; Spärck Jones, 1964; Turney, 2006). Such distributional (or *vector-space*) representations of meaning induce measures of word similarity that can be tuned to correlate well with measurements made by humans.

Much previous work has focused on designing feature representations and semantic spaces that capture salient properties of word meaning (Agirre et al., 2009; Gabrilovich and Markovitch, 2007; Curran, 2004b), directly leveraging the *distributional hypothesis*, i.e. that similar words appear in similar contexts (Harris, 1954; Miller and Charles, 1991; Lin and Pantel, 2002; Pereira et al., 1993). Features are collected form a variety sources, for example: (1) word collocations (Schütze,

1998a), (2) syntactic relations (Padó and Lapata, 2007), (3) structured corpora such as Wikipedia (e.g. Gabrilovich and Markovitch (2007)) or (4) latent semantic spaces (Finkelstein et al., 2001; Landauer and Dumais, 1997; Turian et al., 2010).

Once the feature space has been fixed, similarity between word vectors can be used to address a variety of problems, ranging from modeling human word-association norms (Curran, 2004a; Miller and Charles, 1991; Schütze, 1998a; Turney, 2006), to text classification (Baker and McCallum, 1998), to selectional preference (Resnik, 1997) and lexical substitution (McCarthy and Navigli, 2007).

Using distributional models to capture similarity relations between documents dates back to Salton et al. (1975), who developed a system for document retrieval based on vector-space embedding. User queries are embedded as points in the same space and documents are sorted in order of increasing distance from the query. Due to their scalability there has been significant subsequent work applying distributional lexical semantics to information retrieval (Gorman and Curran, 2006; Manning et al., 2008; Sanderson, 1994),

Erk (2007) introduces a system for computing the selectional preference of semantic roles, marking sentence constituents based on the role they play in relation to the main verb. Pennacchiotti et al. (2008) use a distributional representation to induce the semantic frames for unknown lexical units in a sentence.

Distributional methods have also been applied extensively in *query expansion*: adding additional terms to a search query in order to broaden the potential set of matches. Approaches range from modeling query semantics (Cao et al., 2008),

to mining user session contexts (Huang et al., 2003; Jones et al., 2006), and click contexts, (Wen et al., 2001). Such approaches also are relevant to *computational advertising*, where advertisers buy keywords from search engines in order to target their ads. Several approaches have been proposed in the literature for broadening advertising keyword matches (Chang et al., 2009b; Gleich and Zhukov, 2004).

Finally, distributional methods have been applied to several information extraction and knowledge acquisition tasks such as large-scale taxonomy induction (Snow et al., 2006), attribute extraction (Van Durme and Paşca, 2008) and named entity recognition (Paşca et al., 2006; Vyas et al., 2009).

2.1.1 Distributional Similarity

One of the main results of this thesis is a model-based approach to lexical semantics capable of capturing multiple types of semantic relations. Following Baroni and Lenci (2011), the vague notion of "semantic similarity" can be refined into multiple high level relations:

- **Coordinate** – The degree to which pairs of words have similar meaning; ranging from complete contextual substitutability (absolute synonymy), truth preserving synonymy (propositional synonymy) to near synonymy (plesionymy). Rapp (2003) applied a distributional model of word similarity to Test of English as a Foreign Language (TOEFL) questions (Deerwester et al., 1990), achieving 92.5% accuracy vs. a human baseline of 64.5%. Furthermore, Rapp (2003) was able to demonstrate that context features from a small window around the word outperforms more distant features, at least for determining

synonyms in TOEFL. In similar work, Turney (2006) achieve near-human results on SAT analogies, 56% vs. 57%.

- **Hyponymy** – Or subsumption, the degree to which one word denotes a concept that is a subset of another (e.g. *animals* $\rightarrow$ *tigers*). Large-scale induction of hyponyms from unstructured text has been proposed by Hearst (1992).

- **Hypernymy** – The inverse of hyponymy, i.e. the Is-A or relation (e.g. *people* $\leftarrow$ *architects*). Snow et al. (2006) propose a method for large-scale instrumentation of WordNet with automatically induced hypernyms.

- **Meronym** – Part, or component relation, e.g. *leopards* have *fur*. Reisinger and Paşca (2009) combine WordNet with a hierarchical model of attribute generation in order to extract high precision meronym (attribute) sets.

- **Event** – Noun-verb relations, selectional preference, or "slot-filling." The selectional preference of a verb is the set of nouns that it can act on, for example *hamburgers* can be *eaten* (Resnik, 1997).

Words that share similar collocations are often topically related. For example *football*, *season, fans, games, nfl* are often collocated because they occur commonly in sports related discourse, despite being semantically quite distinct. Such topical lexical relations are said to be **syntagmatic**. In contrast, it is often the case for words to rarely co-occur, but to share contextual usage, e.g. *milkman* and *fireman*; such relations are called **paradigmatic**.

2.1.2 Word Sense Induction

Lexical semantic models are also useful for performing **word sense disambiguation**, i.e. given two occurrence of the same word, along with their contexts, determine whether they refer to the same sense or not (Schütze, 1998a; Agirre et al., 2009; Pedersen and Kulkarni, 2006).

Word sense relations are typically divided into:

- **Homonymy** – Two words that are the same but which denote different meanings (e.g. the river *bank* and the investment *bank*).
- **Polysemy** – Different senses of the same word that share some underlying commonalities (e.g. an academic *journal* and a child's *journal*; or "walk the *line*" vs. "*line* up").

Pereira et al. (1993) induce word senses based on minimum-distortion hierarchical clustering. Latent classes based on these clusters are shown to capture generalizable verb-noun selectional preferences. Pantel and Lin (2002) use soft clustering of words based on features derived from parsed text, and demonstrate that such an approach is capable of discovering the senses of polysemous words. Word sense disambiguation is potentially important for, e.g. understanding search queries (Reisinger and Pasca, 2011), because there is little explicit discourse context and mistakes can potentially cause user frustration.

2.1.3 Contrasts with Resource-based Approaches

Outside of distributional approaches, previous work on lexical semantic relatedness has also focused on mining monolingual or bilingual dictionaries or other pre-existing resources to construct networks of related words (Agirre and Edmonds, 2006; Ramage et al., 2009b). This approach tends to have greater precision, but depends on hand-crafted dictionaries and cannot, in general, model sense frequency (Budanitsky and Hirst, 2006). The vector-space approach is fundamentally more scalable as it does not rely on specific resources and can model corpus-specific sense distributions. However, it can suffer from poor precision, as thematically similar words (e.g., *singer* and *actor*) and antonyms often occur in similar contexts (Lin et al., 2003). Thus, vector-space models are typically posed as identifying thematically *related* words, rather than synonyms (Agirre et al., 2009).

Logical approaches to machine translation (Emele et al., 1996) or semantic parsing (Bayer et al., 2004; Joshi and Vijay-Shanker, 1999; Zettlemoyer and Collins, 2005) historically have required hand-specification of rules and other semantic resources. Hand specified models can be extremely accurate, but are often brittle and inflexible with respect to ambiguity, scaling poorly to larger and more open domains. More recent work overcomes these constraints by learning logical formalism from observational data (e.g. Artzi and Zettlemoyer, 2011; Chen and Mooney, 2011; Liang et al., 2011).

The most comprehensive and influential approach to resource-based lexical semantics is *WordNet*, an electronic resource generalizing and codifying word meaning and relations between words, and inspired by psycholinguistic theories of

human lexical organization (Fellbaum, 1998b). WordNet consists of English nouns, verbs, adjectives and adverbs organised into *synsets* corresponding to individual conceptual units (generalizing word senses). Synsets are linked into overlapping hierarchies (directed acyclic graphs) based on hypernymy and hyponymy.

Although WordNet has found significant use in computational linguistics, the main issues precluding more widespread use of such resource-based approaches is lack of coverage and high-cost of resource construction (due to the involvement of human editors and staff). Distributional approaches have been applied to automate construction of lexical semantic resources (Curran and Moens, 2002; Grefenstette, 1994; Pantel and Lin, 2002; Snow et al., 2006). Furthermore recent advances in crowd-sourcing promise much cheaper and more comprehensive lexical resources, however controlling for quality is not necessarily straightforward (Biemann and Nygaard, 2010; Ma et al., 2009; Snow et al., 2006).

2.1.4 Limitations of Distributional Models

Standard implementations of distributional models fail to capture the richness of word meaning, since similarity is not a globally consistent metric over word-types. In particular it violates both symmetry: e.g. people have the intuition that *North Korea* is more similar to *China* than *China* is to *North Korea* and the triangle inequality: e.g., the sum of distances from *bat* to *club* and *club* to *association* is less than the distance from *bat* to *association* due to ambiguity in the word-type *bat* (Griffiths et al., 2007b; Tversky and Gati, 1982).

Violations of the triangle inequality can be resolved by first breaking up

word types across their component senses (i.e. multiple prototype models Agirre and Edmonds, 2006; Reisinger and Mooney, 2010; Schütze, 1998a), or using exemplar models of meaning (Erk, 2007), which represent words as collections of multiple word occurrences.[1]

Multiple prototype models capture the underlying concept associate with each word sense as an abstract prototypical instance, similar to a cluster centroid in parametric density estimation (Anderson, 1990). Chapter 4 introduces several model-based approaches to representing words with multiple prototypes.

Exemplar models represent concepts by a concrete set of observed instances, similar to nonparametric approaches to density estimation in statistics (Ashby and Alfonso-Reese, 1995). For example, Erk (2007) represents words as multiple exemplars derived directly from word occurrences embedded in a common vector space, and demonstrate how such a model is capable of capturing context-dependent usage.

Voorspoels et al. (2009) demonstrate the superior performance of exemplar models for *concept combination* (e.g. "metal spoon"), suggesting their use in computational lexical semantics when contextual information is available. In general exemplar models are better suited to address polysemy and contextual variation than prototype models. Indeed, experimental evidence suggests that although polysemous words share the same lexical representation, their underlying senses are represented separately, as priming a word in one sense interferes with using it in

[1] Asymmetry, however, can only be resolved through the use of multiple conditional similarity metrics.

another, even when the senses are related (Klein and Murphy, 2001, 2002).

Moving beyond consistency violations, word relatedness for a given pair implicitly defines a typed relation between that pair that may not at all be similar to the relations between similar words. For example *wine* and *bottle* are similar and *wine* and *vinegar* are similar, but it would not be reasonable to expect that the features governing such similarity computations to overlap much, despite all three words occurring in similar documents. The aim of this thesis is to study the application of *cross-categorization* (Smith and Shafto, 2011) to find coherent feature subsets that implicitly define meaningful relations, resulting in vector-valued word relatedness.

The main contribution of this thesis (Chapter 5) is to explore the degree to which the overlapping categorization structure of concepts can account for generalization and variation in word meaning and help overcome feature noise.

2.2 Types of Distributional Models

In this thesis, I will divide distributional lexical semantics models into two categories based on the source data type:

1. *Word Occurrence* models, e.g. exemplar or prototype models, which seek to capture the empirical distribution of individual word occurrences in context. The defining feature of such models is the ability to build representations independently for each word type. Operationally, such models are trained on contextual occurrences of each word-type. Such models will be discussed in

Chapter 4 and include multi-prototype and tiered clustering.

2. *Word Type* models, including most latent variable model-based approaches. These models construct representational vectors at the level of word-types, conflating the contribution of each individual word occurrence (and thus senses). Models are trained over all words in the corpus jointly, potentially allowing latent variables to "pick apart" the conflated vectors and retrieve senses. An example is fitting a dimensionality reduction model such as Latent Dirichlet Allocation (LDA) to a set of word-types and treating each topic as a latent sense. Such models will be discussed in Chapter 5 and include MVM and LDA.

These two classes of models are treated separately in this thesis because the underlying data requirements differ: word type models can be trained on both occurrence-level (such as raw text) and aggregate features (such as the Google n-gram corpora); however, word occurrence models can only be trained on word occurrence data.

At first glance, it would seem that word type models would not be able to capture contextual dependence or ambiguity since they conflate individual occurrence vectors; however, this is indeed not the case, as particular forms of latent variable modeling can reconstruct polysemous usages.

Word-occurrence models are more computationally tractable and can be parallelized naively by word-type. However, since this method treats each word type independently, the usages discovered for w cannot influence the usages discovered for $w' \neq w$. Sharing statistical strength across similar words could yield better re-

sults for rarer words, in addition to providing a more coherent model of human conceptual organization. Furthermore, the joint word-type model automatically computes inter-word similarity, obviating the need for defining similarity metrics on multiple clusterings.

In latent variable word-type models, such as those based on LDA, the latent topics can be used to de-aggregate the underlying context vectors, accounting for polysemy even when multiple senses have been encoded in the same feature vector. For example, when clustering *apple* with other fruits, LDA might find certain features such as *stock* or *company* to be irrelevant, ignoring the homonymous usage.

2.2.1 Word-Occurrence Models

For addressing context-dependence, Schütze (1998a) introduce a *second-order* vector space model where each word in context is represented by a convex combination of the distributional vectors for nearby terms. The first-order vectors are in turn are constructed by combining local bag-of-words features across all word occurrences.

Mitchell and Lapata (2008) argue that such approaches which explicitly ignore the contribution of syntactic structure are impoverished. To address this, they introduce a notion of vector composition in terms of additive and multiplicative functions, capturing the effects of syntactic structure. Although the composition methods they propose are inspired by the effects of syntax, the actual vector composition methods are insensitive to syntactic relations and word-order (Erk and Pado, 2008).

Erk and Padó (2010) introduces a structured vector space model, using a set of occurrence *exemplars* as the underlying representation for each word instead of a single vector. The interpretation of a word in a given context is a combination of the word's meaning vector with the inverse selection preference of of its context, avoiding conflating meaning vectors from nouns and their verbs directly. Such methods are prone to overfitting and noise, however, as individual word occurrences are sparse, and activating too many exemplar introduces irrelevant features.

Thater et al. (2010) introduce a notion of "syntactically informed contextualization" combining the second-order vector representations from Schütze (1998a) with the selectional preference constraints introduced by Erk and Pado (2008). Their distributional vectors are composed of "words typically co-occurring with the contexts in which a word typically appears" (Thater et al., 2010). For example, to derive a representation for "acquire knowledge," their model would combine the first-order vector of "knowledge" with the second-order vector for "acquire" using point-wise multiplication. This operation has the effect of filtering the second-order vector of "acquire," refining the meaning representation.

Higher order compositional vector spaces based on tensor algebra have also been proposed, dating back to Smolensky (1990)(see e.g. Baroni et al., 2010; Baroni and Zamparelli, 2010; Rudolph and Giesbrecht, 2010; Grefenstette et al., 2011); although they allow richer semantics to captured in a uniform way, these methods currently scale poorly with corpus size, as storage overhead increases exponentially with each dimension.

2.2.2 Word-Type Models

Pereira et al. (1993) introduce an approach to word-type distributional lexical semantics based on minimum-distortion hierarchical clustering. Word clusters are used to construct a *class model* for tagging words with their latent semantic class. These classes are shown to capture generalizable verb-noun selectional preferences.

Landauer and Dumais (1997) use Latent Semantic Analysis (LSA) to perform dimensionality reduction on co-occurrence vectors, demonstrating good generalization performance completing English analogies. For contextualization, however, they sum the latent vector representations of each context word, following Schütze (1998a). LSA and its probabilistic extension Latent Dirichlet Allocation (LDA; Blei et al., 2003b) have been demonstrated to yield generalizations that correlate well with human production norms (Griffiths et al., 2007a).

Dinu and Lapata (2010) fit LDA to first-order word co-occurrence vectors, similar to Landauer and Dumais (1997), and then derive probabilistic machinery for combining head word vectors with vectors from their local contexts. The main assumption driving this work is that word meaning can be represented as a probability distribution over a set of latent senses modulated by context. Topic models like LDA have also been used for other lexical semantic tasks, such as word sense induction (Brody and Lapata, 2009; Li et al., 2010; Yao and Durme, 2011).

Rooth et al. (1999) apply latent variable modeling to lexical substitution, modeling slots as joint multinomial distributions over arguments and relations. Sev-

eral authors have recently revisited this model, casting it in the Bayesian LDA framework:

- Ritter et al. (2010) demonstrated significant gains from applying a relational extension of LDA to jointly model the selectional preference of a large subset of TextRunner relations (Banko et al., 2007), e.g.

  André Gide — died in — Paris.

  Latent distributions over target words are coupled with latent distributions over local syntactic relations, yielding a model of verb-slots suitable for modeling selectional preference.

- Ó Séaghdha (2010) derive a substantially similar approach, and train on a much smaller data set. Their results are competitive with the state of the art, especially for infrequent verb-argument pairs.

Ó Séaghdha and Korhonen (2011) extend their previous latent variable work with additional syntactic information, using the probabilistic machinery from Dinu and Lapata (2010) to incorporate local dependency contexts instead of co-occurrence windows.

The related work outlined in this section will constitute the baseline models when empirically evaluating the cross-cutting distributional models. In particular, the probabilistic contextualization methodology introduced in Dinu and Lapata (2010) and extended in Ó Séaghdha and Korhonen (2011) will be generalized and adapted to cross-cutting models. In the next section I will introduce the empirical

evidence for cross-cutting models themselves, in the context of human conceptual organization.

Van de Cruys et al. (2011) introduce a latent variable model based on matrix factorization for capturing word meaning in context. Word types, along with their window-based contexts and dependency relations are linked to latent dimensions, which capture what dimensions are important for a particular context. Evaluation in both English and French shows that the results exceed the previous state-of-the-art for lexical substitution.

Thater et al. (2011) that represents contextualized words based on their local syntactic context. Features in the global word-type vector are reweighted based on distributional information of the context words. This model outperforms previous models on the SemEval 2007 Lexical Substitution task (§3.2.5), achieving state of the art performance.

2.3 Cross-cutting Models of Conceptual Organization

Psychological models of concept categorization have been motivated from a large variety of approaches, including:

- Synthesizing category members into abstract conceptual prototypes (Anderson, 1990; Posner and Keele, 1968),
- Identifying exemplar members for each category (Ashby and Alfonso-Reese, 1995; Kruschke, 1992; Medin and Schaffer, 1978; Nosofsky, 1986),

- Learning categorization rules (Nosofsky et al., 1994; Goodman et al., 2008),
- Achieving parsimony or simple models (Pothos and Chater, 2002; Pothos and Close, 2007),
- Hybrid approaches generalizing prototype and exemplar methods (Anderson, 1990; Love et al., 2004).

In this thesis I focus on models capable of capturing context-dependent *selective attention*, which requires multiple systems of categories. Studies from multiple real-world domains have demonstrated evidence for rich category structure, including multiple cross-cutting categorization systems (Smith and Shafto, 2011).

Cross-categorization of concepts arises due to context-dependence: humans attend to different features of a stimuli in different contexts, and categorize objects differently depending on the set of active features (Smith and Shafto, 2011). Models of selective attention typically account for cross-categorization in a stage-wise manner: concepts are categorized based all available features and then additional models are fit to capture feature variance that is not explained well by the previous models (Love et al., 2004; Medin and Schaffer, 1978; Nosofsky, 1986; Shepard et al., 1961). The result of this process is multiple systems of categories that apply in different contexts. Smith and Shafto (2011) point out that *a priori* fixing feature groups before learning categories is somewhat cumbersome, and instead propose an alternative approach that allows feature subsets and categories to mutually constrain themselves, leading to the formation of more parsimonious categories.

2.3.1 Model Specification

The basic Cross-cat model consists of:

- $d \in [1 \dots D]$ F-dimensional data vectors $\mathbf{w} = [\mathbf{w}_1, \dots, \mathbf{w}_D]^\top$.
- $m \in [1 \dots M]$ views defined by $\mathbf{Z}$. View m is a binary vector specifying which features are included in the mth clustering.
- $k \in [1 \dots K_m]$ clusters in clustering $m \in [1 \dots M]$, $\mathbf{c}_k^m$.

Define the unary *factorial feature projection operator*

$$(\star \mathbf{Z}_{\cdot,m}) : \mathbb{R}^F \rightarrow \mathbb{R}^{||\mathbf{Z}_{\cdot,m}||_1}, \tag{2.1}$$

mapping data vectors of dimension F to vectors with dimension equal to the number of nonzero entries of the column-vector $\mathbf{Z}_{\cdot,m}$ (i.e. $||\mathbf{Z}_{\cdot,m}||_1$). Let

$$\lambda^m \stackrel{\text{def}}{=} \{j : j \in [1 \dots F], [\mathbf{Z}]_{j,m} = 1\} \tag{2.2}$$

be the ordered indices of the nonzero entries of $\mathbf{Z}_{\cdot,m}$ and let $L^m \stackrel{\text{def}}{=} |\lambda^m| = ||\mathbf{Z}_{\cdot,m}||_1$ be the number of nonzero entries. Then define

$$\mathbf{w} \star \mathbf{Z}_{\cdot,m} \stackrel{\text{def}}{=} (w_{\lambda_1^m}, \dots, w_{\lambda_{L^m}^m})^\top, \tag{2.3}$$

i.e. the projection of $\mathbf{w}$ onto the lower-dimensional subspace specified by the nonzero entries of $\mathbf{Z}_{\cdot,m}$. Finally $\mathbf{w}^{\star(m)}$ will be used as shorthand for $\mathbf{w} \star \mathbf{Z}_{\cdot,m}$ when the view assignment matrix $\mathbf{Z}$ is unambiguous.

Cross-cat is defined as

$$P(\mathbf{Z}, \mathbf{c}|\mathbf{w}) \propto P(\mathbf{Z}, \{\mathbf{c}^m\}, \mathbf{w}) \tag{2.4}$$

$$= P(\mathbf{Z}) \prod_{m=1}^{M} P(\mathbf{w}^{\star(m)}|\mathbf{c}^m) P(\mathbf{c}^m). \tag{2.5}$$

where $P(\mathbf{Z})$ is the prior distribution on views and $P(\mathbf{c}^m)$ is a prior on the clustering for view m, e.g. the DPMM, and $P(\mathbf{w}^{\star(m)}|\mathbf{c}^m)$ is the likelihood of the data $\mathbf{w}$ restricted to the feature subset $\mathbf{Z}_{\cdot,m}$ given the corresponding clustering $\mathbf{c}^m$ (Shafto et al., 2006).

$P(\mathbf{Z})$ is constructed by first drawing the vector $\tilde{\mathbf{z}} \sim \text{CRP}(\alpha)$, i.e. assigning each feature to some view via the *Chinese Restaurant Process* (Pitman, 1995). $\mathbf{Z}$ is then derived from $\tilde{\mathbf{z}}$ in the obvious way: each feature (row vector of $\mathbf{Z}$) has only a single nonzero entry corresponding to the column index of the view it is assigned to via $\tilde{\mathbf{z}}$:

$$[\mathbf{Z}]_{f,m} = \begin{cases} 1 & \tilde{\mathbf{z}}_f = m, \\ 0 & \text{otherwise.} \end{cases} \tag{2.6}$$

The cross-cat model is capable of finding disjoint views with maximally probable clusterings. The Dirichlet Process parameter α on view assignment controls the trade-off between the fit of any one clustering and the cost of adding an addition clustering, taking features away from the others. Because views form hard partitions of features, cross-cat is not capable of representing *all* of the most probable clusterings simultaneously, i.e. features cannot be shared across views. The Multi-view clustering model (MV-C) introduced in Chapter 5 addresses exactly this issue.

2.3.2 Evidence for Cross-cutting Categorization

Humans use overlapping taxonomies to organize conceptual information in many domains (Ross and Murphy, 1999); i.e. foods can be organized situationally, *breakfast food*, *dinner food*, *snack*, etc, or by their type, *dairy*, *meat*, etc. Each organization system may have different salient features and hence yield different patterns of similarity generalization (cf. Heit and Rubinstein, 1994). For example, Shafto and Coley (2003) find that when reasoning about the anatomical properties of animals relies on taxonomic categories such as *mammals* or *reptiles* whereas reasoning about disease transmission relies on ecological categories such as *predator* and *prey*.

Several studies provide experimental evidence for multiple categorization across a number of different domains. Ross and Murphy (1999) demonstrate peoples' use of both taxonomic (e.g. grain) and situational categories (e.g. eaten for breakfast) when make inferences about food. In particular, when presented with individual foods and asked to categorize them, people list $\approx$50% taxonomic categories and $\approx$42% situational categories. Furthermore, both situational and taxonomic category labels prime retrieval of category members and guide inductive inference.

Building on previous work in folk-taxonomic categorization in biology (Boster and Johnson, 1989; Medin et al., 2006), Shafto and Coley (2003) find evidence for multiple categorization systems in marine creatures. Novices were found to classify fish based on appearance, while experts relied more on ecological niche.

In related work, Heit and Rubinstein (1994) demonstrate that people use taxonomic knowledge when reasoning about anatomical properties, but use ecological knowledge when reasoning about behavioral properties. They draw the conclusion that:

> prior knowledge could be used dynamically to focus on certain features when similarity is evaluated. In this conception, inductive reasoning is an active process in which people identify features in the premise and conclusion categories that are relevant to the property being inferred.

i.e. that people focus on specific, not general, ways in which concepts are related (Murphy, 2002). Conceptual similarity is determined contextually and inference is relative to a subset of active features, not the total set of available features.

In studying amateur and expert physics students, Chi et al. (1981) provide evidence for multiple categorization systems: expert students augment similarity-based event categories with categories based on abstract physical principles, and these abstract categories play an important role in expert problem solving.

Multiple organizational systems are also used in categorical models of social perception, i.e. people can be categorized by gender, age, race, or occupation; further, categories toward which people have highly accessible attitudes (i.e. have strongest polarity sentiment) are preferentially applied to multiply categorizable objects (Smith et al., 1996). Zarate and Smith (1990) demonstrate gender and racial differences in categorization speed when faced with overlapping social categories, indicating a potential biological basis for category priming. Nelson and

Miller (1995) demonstrate similar results for a more general class of distinctive traits, showing differences in feature salience in the perception of social similarity.

Finally, Cross-categorization is also apparent in goal-driven behavior, where categories may be derived on-the-fly while planning (Barsalou, 1991). For example, when planning for a vacation in SF, people may form the categories "departure times that minimize work disruption", "people who live in California", and "things to pack in a small suitcase."

Chapter 3

Evaluating Lexical Semantic Models

This chapter introduces the textual corpora used to train the various lexical semantics models (§3.1), and describes the set of evaluation tasks employed in the subsequent chapters (§3.2).

3.1 Corpora and Features

Models introduced in the subsequent chapters can be divided into two groups based on the way in which the base data is used:

- Models based on **word-occurrence**, e.g. multi-prototype and tiered clustering, where each unique occurrence of a word-type in context is included as a separate data point; and
- Models based on **word-type** features, e.g. MVM, where word-type feature vectors are collapsed across all occurrences, yielding one data point per word type.

Word vectors are derived from four corpora: (1) the Google Web n-gram corpus (word-type), (2) the Google Books n-gram corpus (word-type), (3) a 10/2010 snapshot of the English Wikipedia (word-occurence) and (4) the English Gigaword cor-

pus (word-occurrence). Features are collected for 42K target words ranked by absolute term frequency in the combined Google corpora.

3.1.1 Word-Occurrence Features

The Multi-prototype and tiered clustering models assign individual word occurrences in context to clusters representing their sense. The base data for these models is raw bag-of-words occurrence data for a set of target head words (see Table 3.1 for an example from Wikipedia).

Word occurrence data is derived from two corpora:

1. **Wikipedia** – A snapshot of English Wikipedia taken on October. 11th, 2010. Wikitext markup is removed, as are articles with fewer than 100 words, leaving 2.8M articles with a total of 2.05B words.

2. **Gigaword** – The third edition English Gigaword corpus, with articles containing fewer than 100 words removed, leaving 6.6M articles and 3.9B words (Graff, 2003).

Wikipedia covers a wider range of sense distributions, whereas Gigaword contains only news-wire text and tends to employ fewer senses of most ambiguous words.

Head Word	Features
board	on all Baptie people Charles All Airlines four
can	Of accurate but historical hindsight aid notability more
current	and form Spoken of literary including previous
elected	and of Strategy as Marsh Chairman Doug steering Resources
following	and Ashley turn see Laurel's Doug Andrea
game	good forage fair mule as rates wild such
give	gist don't of separate but needed done Related think
have	use philosophy that of regional commitment committed such impose government's
horse	eventually use be wooden Greeks construct proposes of on left
include	picnic that of modern sewer variety campsites
long	doesn't use anything prefer but actually break as inherit
over	blue triumphed Award present crab team red
problem	reset full ways of There this various fixing Doing
release	on After takes material long time new with
slightly	be mainspace work should edits want you might
talk	on page Jul Sat
that	maybe move Merge with correct Isn't
time	and constitutions provisions of could draft new meet until
type	scaled essentially that of value fixed-point have specific data by
under	and present

Table 3.1: Example head words and bag-of-words features for a single occurrence, collected using Wikipedia.

3.1.2 Word-Type Features

In order to test their scalability, the word-type models (e.g. MVM) are trained on a combination of the publicly available Google Books n-gram data[1], the Google Web 5-gram data available from LDC,[2] and Wikipedia article occurrence counts.

The Google corpora consist of n-gram contexts and associated frequency counts, e.g.:

not get a chance to	137788
actually have a chance of	1890
ya get a chance	3071
cloudy with a chance	317793

Two types are features are extracted from this data:

1. **Context Features** – All n-gram contexts containing an instance of a target word are collected. These contexts are then converted into general *slots* in which the target word has been removed. E.g., for the target *chance* and the context set above the following features are generated:

cloudy with a __	317793
not get a __ to	137788
ya get a __	3071
actually have a __ of	1890

[1] The Google books data contains additional yearly frequency data; we sum all occurrences over all years. http://books.google.com/ngrams

[2] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2006T13

2. **Unigram Co-occurrence Features** – In addition to context features, unigrams appearing in the same context window (up to 5-gram) as a target word are also collected. E.g., for the target *chance* and the context set above, the following features are generated:

a	458652
cloudy	317793
with	317793
get	140859
not	137788
to	137788
ya	3071
actually	1890
have	1890

In addition, Wikipedia **article occurrence** counts are collected for each head word, corresponding to a semantic as opposed to syntactic context. Article occurrence counts are derived simply from counting the number of times a head word appears in each article. For example, the headword *chance* contains features such as

Infinite monkey theorem	18
Houston Astros	7
North Melbourne Football Club	5
Cousin marriage	4
Charles Sanders Peirce	4
African-American Civil Rights Movement (1955-1968)	4
Starscream (Transformers)	4
Star Trek VI: The Undiscovered Country	4

English Wikipedia articles with fewer than 1000 words or less than 5 incoming links are discarded, leaving a total of 113K documents.

Table 3.2 breaks down the total number of features collected across all 42K target words. In the final base data, each target word was limited to a maximum

source	feature type	unique count
Google Web	2-gram context	3.9M
Google Books	2-gram context	1.3M
Google Web	3-gram context	118.7M
Google Books	3-gram context	38.4M
Google Web	4-gram context	431.2M
Google Books	4-gram context	160.8M
Google Web	5-gram context	651.4M
Google Books	5-gram context	244.3M
Google Web	unigram co-occurrence	1.7M
Google Books	unigram co-occurrence	362K
Wikipedia	article occurrence	113K

Table 3.2: Unique feature counts for the word-type data broken down across feature type and source corpus.

of 20k unique features, ranked by *t-test* score. Furthermore, features co-occurring with less than 100 target head words were discarded (for Wikipedia article features this limit was set to 4 head words).

3.2 Lexical Semantic Tasks

This section describes the five lexical semantics tasks used to evaluate the proposed models.

3.2.1 Semantic Similarity

The most basic approach to evaluating distributional lexical semantic models is comparing their predicted pairwise similarity scores to a set of similarity measurements collected from human raters. In this thesis I make use of two such

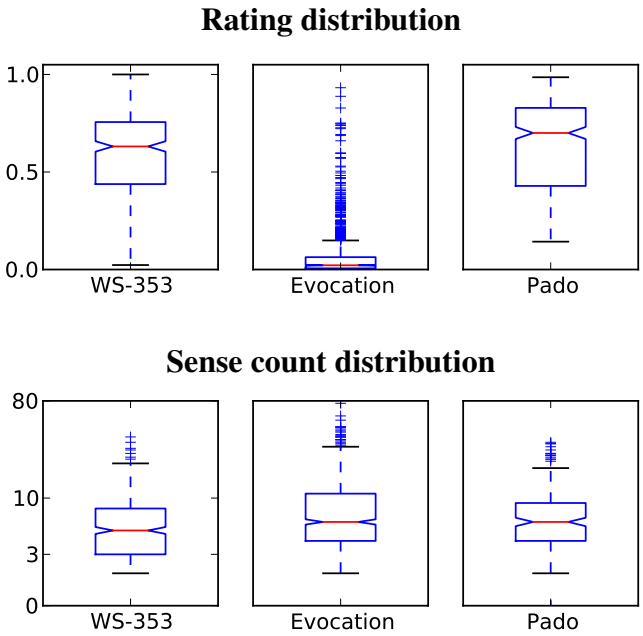


Figure 3.1: (**top**) The distribution of ratings (scaled [0,1]) on WS-353, WN-Evocation and Padó datasets. (**bottom**) The distribution of sense counts for each data set (log-domain), collected from WordNet 3.0.

test collections:

1. **WS-353** contains between 13 and 16 human similarity judgements for each of 353 word pairs, rated on a 1–10 integer scale (Finkelstein et al., 2001).[3]

2. The Princeton **WN-Evocation** contains over 100k similarity comparisons collected from both trained human raters (WN-Evocation-Controlled) and

[3](**Similarity vs. Relatedness**) One issue with measuring semantic similarity is that it conflates various types of relations, e.g. hyponymy, synonymy or metonymy. In order to better analyze the various components of attributional similarity, Agirre et al. (2009) divide the WS-353 dataset into separate *similarity* and *relatedness* judgements. Similar pairs include synonyms, antonyms and hyponym-hypernyms; related pairs consist of meronym-holonyms and others that do not fit the previous relations. The analyses presented here are extended to these subsets.

participants on Amazon's Mechanical Turk (WN-Evocation-MT; Ma et al., 2009). WN-Evocation comparisons are assigned to only 3-5 human raters on average and contain a significant fraction of zero- and low-similarity items compared to WS-353 (Figure 3.1), reflecting more accurately real-world lexical semantics settings. In these experiments I discard all comparisons with fewer than 5 ratings and then sample 10% of the remaining pairs uniformly at random, resulting in a test set with 1317 comparisons.

Evaluation: Each model is used to generate a similarity measures between each pair of words in the corpus. **Spearman's Rank Correlation Coefficient** (Spearman's ρ) is then used to measure the correlation between the human ratings and the model-produced ratings: Given a list of items w with gold standard ranks g_w and model-produced ranks m_w, *Spearman's* nonparametric rank correlation coefficient is defined as

$$\rho = \frac{\sum_w (m_w - \bar{m})(g_w - \bar{g})}{\sqrt{\sum_w (m_w - \bar{m})^2 (g_w - \bar{g})^2}}, \tag{3.1}$$

i.e., Pearson's ρ over the ranks of items (Agirre et al., 2009).

3.2.2 Selectional Preference

Selectional preference is the task of predicting the typical filler of an argument slot of a verb (Resnik, 1997; Pantel et al., 2007); e.g., the set of things that can be *eaten* or *thrown*. Distributional methods have proven to be a powerful approach to modeling *selectional preference* (Padó et al., 2007; Pantel et al., 2007), rivaling methods based on existing semantic resources such as WordNet (Clark and Weir, 2002; Resnik, 1997) and FrameNet (Padó, 2007) and performing nearly as well as

supervised methods (Herdağdelen and Baroni, 2009). Selectional preference has been shown to be useful for, e.g., resolving ambiguous attachments (Hindle and Rooth, 1991), word sense disambiguation (McCarthy and Carroll, 2003b) and semantic role labeling (Gildea and Jurafsky, 2002).

In order to evaluate selectional preference models, I employ the *Padó* dataset, which contains 211 verb-noun pairs with human similarity judgements for how plausible the noun is for each argument of the verb (2 arguments per verb, corresponding roughly to subject and object; see Table 3.3). Results are averaged across 20 raters; typical inter-rater agreement is $\rho = 0.7$ (Padó et al., 2007).

Evaluation: Each model is used to generate a similarity measures between each verb-noun pair in the corpus. *Spearman's* ρ is then used to measure the correlation between the human argument typicality ratings and the model-produced ratings (Equation 3.1).

3.2.3 McRae Typicality Norms

Models of conceptual organization are ultimately grounded in some feature space as with vector-space lexical semantic models. Semantic feature production norms—i.e., what features people most often report as salient to a given stimulus concept— have been studied extensively in psychology as one way to understand human concept organization and categorization. McRae et al. (2005) introduce a large, balanced set of human feature production norms covering 541 living and non-living things in order to study concept organization and categorization. This data set consists of class-attribute pairs, e.g. *reptile* and *lives in jungle*, or *device*

Verb	Slot	Noun	Typicality
hear	agent	ear	6.8
ask	patient	doctor	6.7
ask	agent	prosecutor	6.6
ask	patient	police	6.2
advise	agent	designer	5.8
hit	agent	opponent	5.3
tell	patient	department	5.0
caution	patient	friend	5.0
embarrass	patient	executive	4.9
raise	agent	question	4.2
encourage	agent	company	4.0
advise	patient	doctor	4.0
see	patient	viewer	3.8
see	patient	drop	3.8
inform	agent	public	3.8
confuse	agent	shareholder	3.3
increase	patient	industry	3.0
resent	agent	transfer	1.3
embarrass	patient	revelation	1.2
eat	patient	group	1.1

Table 3.3: **Padó Selectional Preference Dataset**: Example verb-noun pairs and associated typicality scores. Nouns are associated with either the agent or patient slot of the verb.

category	exemplar	typicality
fruit	plum	6.5
utensil	ladle	6.2
storage	shelves	6.0
mammal	leopard	6.0
vehicle	bus	5.6
mammal	hamster	5.6
fish	goldfish	5.6
rodent	chipmunk	5.5
thing	doll	5.3
building	cabin	4.8
utensil	whip	4.7
tools	blender	4.6
bird	flamingo	4.6
storage	bottle	4.4
device	couch	4.2
weapon	crossbow	3.8
tools	tomahawk	3.7
animal	walrus	3.2
housing	beehive	3.1
tools	stereo	2.6
animal	prune	1.9

Table 3.4: **McRae Typicality Norms**: Examples of category labels and associated exemplars ranked by typicality score.

and *contains machinery*.

Fountain and Lapata (2010) extend this data set with category typicality information, e.g. when asked to give examples of the class *reptile*, humans are more likely to respond with *rattlesnake* than *leopard*. Participants on Amazon's Mechanical Turk were presented with concepts and asked to label them with the most appropriate category (free-form), yielding 541 exemplars for 41 total categories. A

second set of participants were asked to numerically gauge the typicality of each category-exemplar pair. Sample typicality pairs and scores are shown in Table 3.4.[4]

Evaluation: The performance of each model is evaluated on the McRae typicality data set using three tasks:

1. **Category naming** – Can the concept representation for each exemplar be used to predict its category label? All concept labels from the entire set are pooled and ranked by similarity score to the target exemplar. Performance is summarized by gold category label **Recall**:

   Given a list of ranked results r and an unranked set of gold queries q, the *recall at* n (R_n) is computed as:

   $$R_n(r, q) \stackrel{\text{def}}{=} \frac{|\{r_i | i < n\} \cap q|}{|q|}, \tag{3.2}$$

   where r_i is the ith ranked result in r and $|\{r_i | i < n\} \cap q|$ is the number of elements in q also in the first n elements of r.

   and **Mean Reciprocal Rank**:

   Given a list of ranked results r and an unranked set of gold queries q, the *Mean Reciprocal Rank* (MRR) score is computed as:

   $$\text{MRR}(r, q) = \frac{1}{|q|} \sum_{q_i \in q} \frac{1}{\text{rank}(q_i, r)}, \tag{3.3}$$

   where $\text{rank}(q_i, r)$ is the rank of the ith gold query in r (or zero if $i \notin r$).

[4] http://homepages.inf.ed.ac.uk/s0897549/data/

2. **Exemplar generation** – Can the concept representation be used to predict additional (unseen) exemplars? All exemplars across all categories are pooled together as an in-domain set and ranked based on their semantic similarity to the concept. The model's ability to produce correct exemplars is summarized using recall and MRR (Equation 3.3).

For each evaluation, two concept representations are considered:[5]

1. **Category Label** – The concept is represented by the feature vector corresponding to its label (e.g. the concept *bird* is represented by the distributional vector for the word *bird*).

2. **Cateory Constituent** – The concept is represented by the *typicality*-weighted centroid of the feature vectors of its constituent exemplars; e.g. the concept *bird* is represented by the weighted vector centroid of *falcon*, *oriole*, etc. For the exemplar generation, the target exemplar is removed before performing the weighted average.

3.2.4 Baroni and Lenci Evaluation of Semantic Spaces

Baroni and Lenci (2011) introduce a unified data set for systematic, intrinsic evaluation of semantic spaces (the *Baroni and Lenci Evaluation of Semantic Spaces*; **BLESS**).[6] In particular, the goal of the BLESS task is to generalize several

[5] In the original paper, Fountain and Lapata (2010) term these *prototype* and *exemplar* concept representations. However, such double usage of the terms is confusing in this context, and I adopt a different set of names.

[6] http://clic.cimec.unitn.it/distsem

concept	class	relation	argument
ant-n	insect	mero	antenna-n
cabbage-n	vegetable	mero	head-n
cannon-n	weapon	attri	large-j
car-n	vehicle	mero	brake-n
castle-n	building	mero	furniture-n
cat-n	ground_mammal	coord	rabbit-n
cello-n	musical_instrument	event	practice-v
dress-n	clothing	coord	shirt-n
fox-n	ground_mammal	mero	mouth-n
hospital-n	building	mero	doctor-n
knife-n	tool	attri	old-j
lime-n	fruit	attri	sour-j
onion-n	vegetable	coord	leek-n
owl-n	bird	event	inhabit-v
sofa-n	furniture	attri	comfortable-j
tiger-n	ground_mammal	mero	claw-n
trumpet-n	musical_instrument	coord	harmonica-n
trumpet-n	musical_instrument	event	listen-v
yacht-n	vehicle	event	leave-v
yacht-n	vehicle	mero	fin-n

Table 3.5: (**BLESS**) Examples of relations between concepts and arguments, including the concept class.

commonly studied semantic similarity tasks, e.g. **WS-353** (§3.2.1; Finkelstein et al. (2001)) and the **McRae** production norms (§3.2.3; McRae et al. (2005)).

The BLESS data set consists of 200 distinct English concrete nouns divided into 17 categories: **amphibian-reptile**, **appliance**, **bird**, **building**, **clothing**, **container**, **fruit**, **furniture**, **ground-mammal**, **insect**, **musical-instrument**, **tool**, **tree**, **vegetable**, **vehicle**, **water-animal**, and **weapon** (with on average 11.76 concepts per class). Related terms are collected for each concept noun following one of five relations:

1. **coord** – The related word is a coordinate term or co-hyponym; i.e. the two concepts belong to the same semantic class (e.g. *alligator* and *lizard*).

2. **hyper** – The related word is a hypernym of the concept, i.e. is a more semantically broad class (e.g. *alligator* and *animal*).

3. **mero** – The related word is a meronym of the concept, i.e. is a part, component or member of the concept (e.g. *alligator* and *mouth*).

4. **attrib** – The related word is an attribute of the concept (e.g. *alligator* and *aquatic*).

5. **event** – The related word is an activity, action or event related to the concept (e.g. *alligator* and *swim*).

Table 3.5 lists 20 example relations. The stated goal of the task is to provide a common platform for comparing semantic spaces and models, in particular eliciting differences in their predictive performance across the five relation types.

Evaluation: All target attributes for a particular concept are collected into an *in-domain* set $\{D_c\}$ and augmented with an additional 60 *random* terms to act as confounders. Recall and MRR (Equation 3.3) are then computed for each concept-relation pair (c, r), yielding insight into which relations are preferred by which models.

3.2.5 Lexical Substitution

Lexical substitution is the task of identifying suitable replacements for a target word in a given sentence. For example in the sentence

> Vincent van Gogh in 1880 was a 27-year-old failure: a despised and rejected clergyman in a **grim** backwater mining town in Belgium.

human evaluators may suggest that the head word *grim* can be replaced with *gloomy*, *harsh*, *horrid*, *depressing* or *bleak*. In the sentence

> When Time magazine's cover portrays millennium nuts as deranged, crazy Christians holding a **cross** as it did last month, boycott their magazine and the products it advertises.

however, evaluators may only suggest one potential substitution: *crucifix*. Potential substitutes can be collected from expert human evaluators (such as in the *LexSub07* dataset described below) or crowdsourced from sites like Amazon's Mechnical Turk (as in the *TWSI1* dataset).

3.2.5.1 Datasets

Two collections of lexical substitution problems are considered in this thesis:

1. **LexSub07** – The English Lexical Substitution task (part of SemEval 2007) consists of 2010 sentences over 205 unique target words annotated for contextual paraphrases (McCarthy and Navigli, 2007). See Table 3.6 for example sentences and potential substitutes.

2. **TWSI1** – The Turk bootstrap Word Sense Inventory (TWSI) consists of 51.8k occurrences of 397 frequent target nouns drawn from Wikipedia. Each occurrence is sense-labeled and each sense is paired with plausible substitutions collected using a Mechanical Turk-based bootstrapping technique (Biemann and Nygaard, 2010). Unlike WordNet, the TWSI sense inventory is determined by common substitutions rather than psychologically motivated concepts. See Table 3.7 for example sentences and word sense substitutes. For evaluation purposes, a random 10% sample of the TWSI data (5191 sentences) is used.

In both tasks, potential replacement words for each head word are ranked by a numeric score determined by human evaluators. The LexSub07 data set contains ranked replacements unique for each context, while TWSI contains replacements only for each word sense. Furthermore, TWSI only contains concrete nouns as target words, while LexSub07 contains a mix of different parts of speech.

Following previous work, we focused on the subtask of ranking contextual substitutes drawn from the closed vocabulary of human annotations with multi-word substitutions removed (Dinu and Lapata, 2010; Ó Séaghdha and Korhonen, 2011; Thater et al., 2010).

Evaluation: Models are scored using **Generalized Average Precision** (GAP; Kishida, 2005), a robust measure of overlap between ranked lists based on average precision:

> $G = ((g_1, a_1), \ldots, (g_m, a_m))$ is a list of gold paraphrases and associated weights for a given sentence. $P = ((p_1, y_1, b_1), \ldots, (p_n, y_n, b_n))$ is a list of model predictions ranked by model score y_i, with associated *gold* weights b_i ($b_i = 0$ if $p_i \notin G$).
>
> Let $I(p_i) = 1$ if $p_i \in G$, zero otherwise. $\overline{b_i} \stackrel{\text{def}}{=} \sum_{k=1}^{i} \frac{b_k}{i}$ is the average gold weight of the first i model predictions, and likewise $\overline{a_i} \stackrel{\text{def}}{=} \sum_{k=1}^{i} \frac{a_k}{i}$ for the first i gold predictions. GAP is defined as
>
> $$\text{GAP}(G, P) \stackrel{\text{def}}{=} \frac{\sum_{i=1}^{n} I(p_i)\overline{b_i}}{\sum_{j=1}^{m} I(g_j)\overline{a_j}}. \tag{3.4}$$

GAP is sensitive to the absolute ranked order of the candidate results unlike average precision, which is only sensitive to relative rank.

id	context	replacements
211	Galls indeed arise from the stinging of the plant tissues by the ovipositors of female gall wasps, and the egg laid in the plant tissues develops inside the gall into a grub, which eventually emerges full-grown and transformed into a mature **gall** wasp.	pn (2), secretion producing (1)
217	And the morans have the **gall** to ruin Beethoven's 6th in the process, too.	cheek (3), audacity (2), nerve (2), effrontery (1), temerity (1)
233	I **lie** down on my futon -bed with a tiny bean-filled pillow under my neck.	stretch out (2), rest (1), recline horizontally (1), get (1), recline (1)
238	Grabbed the blaster **lying** on the seat next to her and fired up at him.	resting (2), sit (1), that was (1), place (1), position (1)
236	"You can't **lie** in front of the bulldozer indefinitely" I'm game... "Zaphod, you look good	recline (2), stay (2), recline horizontally (1), stretch out (1)
239	Whether it's lying through omission, **lying** through misdirection, or outright lies, it's awfully hard to extract nuggets of truth from the noise.	fib (2), fibbing (1), falsify (1), telling untruths (1), telling falsehoods (1), deceive (1)
248	I would like a big, **nasty**, mean, ugly automatic rifle or grenade launcher that I could fire at any car whose alarm blares repeatedly.	mean (2), vicious (1), dirty (1), unpleasant (1), formidable (1), horrible (1), dangerous (1)
256	I hadn't **nearly** finished the work at hand.	almost (2), anyway (1), anywhere near (1), remotely (1), even approximately (1)
294	Chapters from this book by Paul Collins have **run** in McSweeney's Quarterly Concern.	appear (3), appear in sequence (1), be serialised (1), be published (1)
296	Anyway here are some static pictures of the boat - it is driven by a brushless motor driving a prop - but when it'ss **running** you can't see it.	operate (2), go (2), function (1), work (1)
1731	The worm could gave done **truly** random IP generation and that would have allowed it to infect many more systems much faster.	genuinely (4), strictly (1), unequivocally (1), actually (1)
1767	We have not **yet** found any MUD/MOO environments that handle NL processing.	so far (1), hitherto (1), to date (1), until now (1), thus far (1), still (1)
1771	You can now deliver a heavy right-handed **blow** with your fist upon his chin, or over his heart, which will render him unconscious.	hit (2), strike (1), knock (1), setback (1), criticism (1), punch (1), reverse (1)

Table 3.6: Example contextual occurrences and substitutions from the **LexSub07** task.

context	replacements
It dispenses with plot and unlike her earlier work, with its highly specific stage directions, gives no indication what **actions**, if any, the actors should perform on stage, nor does it give any setting for the play.	activity (16), movement (16), motion (12), force (6), behavior (5), maneuver (4), process (4), act (3), function (3), battle (2), conducted activity (2), energy (2), exercise (2), military action (2), operation (2), power (2), service (2)
Violence struck the **campaign** almost as soon as it started.	movement (24), crusade (21), push (16), fight (12), drive (10), operation (8), contest (6), strategy (5), election (4), battle plan (3), election strategy (3), military action (3), plan (3), political campaign (3)
From 1380 to 1382 Louis served as regent for his nephew, King Charles VI of France, but left France in the latter year to claim the throne of Naples following the **death** of Queen Joanna I.	demise (97), passing (84), end (34), decease (22), fatality (15), expiration (14), passing away (14), termination (13), departure (11), dying (11), mortality (11)
9 Any attack in an officer ' s **field** of command triggers a full mobilization, which shall be done in the swiftest possible manner.	area (19), range (11), scope (10), compass (7), space (5), reach (4), force field (3), region (3), ground(s) (2), site (2), span (2), specialty (2), territory (2)
Together with Earth ' s orbital motion of 18 miles per second (29 km per second) speeds can reach 44 miles per second (71 kilometers per second) in **head** - on collisions.	skull (6), crown (4), noggin (4), scalp (4), cranium (2), top (2)
We are not left with many options, ä high - ranking defense **official** told The Jerusalem Post on Tuesday.	officer (11), authority (4), leader (4), executive (3), office holder (3), personage (3), administrator (2), government official (2)
There are other combinations depending on the vacuum **quality** desired.	caliber (11), excellence (11), quality level (11), standard (11), value (9), superiority (6), class (4), grade (4), workmanship (4), worth (4)
In 1054 he was sent to the emperor Henry II to obtain that monarch ' s influence in securing the return to England of Edward the Exile, **son** of Edmund Ironside, who was in Hungary with King Andrew I.	child (59), boy (52), male child (43), offspring (43), heir (23), male offspring (22), kid (13), descendant (10), male heir (10), male progeny (10)

Table 3.7: Example contextual occurrences and substitutions from the **TWSI1** task. Head words are denoted in **bold**, potential substitutes are ranked by human raters for each sense.

Chapter 4

Multi-Prototype Models via Contextual Clustering

4.1 Introduction

Traditionally, word meaning is represented by a single vector of contextual features derived from co-occurrence information, and semantic similarity is computed using some measure of vector distance (Lee, 1999; Lowe, 2001). However, due to homonymy and polysemy, capturing the semantics of a word with a single vector is problematic. For example, the word *club* is similar to both *bat* and *association*, which are not at all similar to each other. Furthermore, most vector-space models are context independent, while the meaning of a word clearly depends on context. The word *club* in "The caveman picked up the *club*" is similar to *bat* in "John hit the robber with a *bat*," but not in "The *bat* flew out of the cave." Formally, these problems arise because word meaning violates the triangle inequality when viewed at the level of word types (Tversky and Gati, 1982). A single "prototype" vector is simply incapable of capturing phenomena such as homonymy and polysemy.

This section describes the **multi-prototype** and **tiered clustering** models, which enrich the standard vector-space word representations, allowing multiple "sense specific" vectors.

4.1.1 Multi-prototype Model

Since vector-space representations are constructed at the lexical level, they conflate multiple word meanings into the same vector, e.g. collapsing occurrences of $bank_{\text{institution}}$ and $bank_{\text{river}}$. *Multi-prototype* representations address this issue by clustering the contexts in which words occur (similar to unsupervised *word sense discovery* (Schütze, 1998a)) and then building meaning vectors from the disambiguated words. More specifically:

- First, a word's occurrence contexts are clustered to produce groups of similar context vectors.
- An average "prototype" vector is then computed separately for each cluster, producing a set of vectors for each word.
- Finally, these cluster vectors can be used to determine the semantic similarity of both isolated words and words in context.

The approach is completely modular, and can integrate any clustering method with any traditional vector-space model. Two variants are explored: (1) a finite clustering version based on the mixture of von Mises-Fisher distributions (§4.2.1), and (2) an infinite version based on the Dirichlet Process Mixture (§4.2.2).

4.1.2 Tiered Clustering Model

Tiered clustering extends the multi-prototype approach with the ability to account for varying degrees of shared (context-independent) feature structure. Although the multi-prototype model can readily capture the structure of homonymous

LIFE
my, you, real, about, your, would
years, spent, rest, lived, last
sentenced, imprisonment, sentence, prison
years, cycle, life, all, expectancy, other
all, life, way, people, human, social, many
RADIO
station, FM, broadcasting, format, AM
radio, station, stations, amateur,
show, station, host, program, radio
stations, song, single, released, airplay
station, operator, radio, equipment, contact
WIZARD
evil, magic, powerful, named, world
Merlin, King, Arthur, powerful, court
spells, magic, cast, wizard, spell, witch
Harry, Dresden, series, Potter, character
STOCK
market, price, stock, company, value, crash
housing, breeding, all, large, stock, many
car, racing, company, cars, summer, NASCAR
stock, extended, folded, card, barrel, cards
rolling, locomotives, new, character, line

Table 4.1: Example DPMM multi-prototype representation of words with varying degrees of polysemy. Each line represents the most common features associated with an inferred word sense. Compared to the tiered clustering results in Table 4.2 the multi-prototype clusters are significantly less pure for *thematically polysemous* words such as *radio* and *wizard*.

words with several unrelated meanings (e.g. *bat* and *club*), it is not suitable for representing the common metaphor structure found in highly polysemous words such as *line* or *run*.

In order to address this problem, I introduce *tiered clustering*, a novel probabilistic model of the *shared structure* often neglected in clustering problems. Tiered clustering performs *soft* feature selection, allocating features between a Dirichlet Process clustering model and a background model consisting of a single component. The background model accounts for features commonly shared by all occurrences (i.e. context-independent feature variation), while the clustering model accounts for variation in word usage (i.e. context-dependent variation, or *word senses*; Table 4.2).

Common tasks in lexical semantics such as word relatedness or selectional preference can benefit from modeling shared structure: Polysemous word usage is often governed by some common background metaphoric usage (e.g. the senses of *line* or *run*), and likewise modeling the selectional preference of verbs relies on identifying commonalities shared by their typical arguments. Tiered clustering can also be viewed as a form of soft feature selection, where features that do not contribute meaningfully to the clustering can be excluded. We demonstrate the applicability of tiered clustering, highlighting particular cases where modeling shared structure is beneficial and where it can be detrimental.

LIFE
all, about, life, would, death
my, you, real, your, about
spent, years, rest, lived, last
sentenced, imprisonment, sentence, prison
insurance, peer, Baron, member, company
Guru, Rabbi, Baba, la, teachings

RADIO
station, radio, stations, television
amateur, frequency, waves, system
show, host, personality, American
song, single, released, airplay
operator, contact, communications, message

WIZARD
evil, powerful, magic, wizard
Merlin, King, Arthur, Arthurian
fairy, wicked, scene, tale
Harry, Potter, Voldemort, Dumbledore

STOCK
stock, all, other, company, new
market, crash, markets, price, prices
housing, breeding, fish, water, horses
car, racing, cars, NASCAR, race, engine
card, cards, player, pile, game, paper
rolling, locomotives, line, new, railway

Table 4.2: Example tiered clustering representation of words with varying degrees of polysemy. Each boxed set shows the most common background (shared) features (top line), and each additional line lists the top features of an inferred prototype vector. Features are depicted ordered by their posterior probability in the trained model given the target word and cluster id. For example, *wizard* is broken up into a background cluster describing features common to all usages of the word (e.g., *magic* and *evil*) and several genre-specific usages (e.g. *Merlin*, *fairy tales* and *Harry Potter*).

4.1.3 Evaluations

The multi-prototype and tiered clustering models are evaluated on a collection of lexical semantic tasks:

- **Paraphrase** – Predicting the most similar words to a given target, both with and without sentential context. The results demonstrate the superiority of a clustered approach over both traditional prototype and exemplar-based vector-space models. For example, given the isolated target word *singer* the multi-prototype method produces the most similar word *vocalist*, while using a single prototype gives *musician*. Given the word *cell* in the context: "The book was published while Piasecki was still in prison, and a copy was delivered to his *cell*." the standard approach produces *protein* while the multi-prototype method yields *incarcerated*.

- **Semantic Similarity** – Two test collections: **WS-353**, which consists of 353 word pairs each with 13-16 human similarity judgements (Finkelstein et al., 2001) and **WN-Evocation**, which contains over 100k similarity comparisons over a much wider vocabulary (§3.2). When combined with aggressive feature pruning, the multi-prototype approach outperforms state-of-the-art vector space models such as Explicit Semantic Analysis (Gabrilovich and Markovitch, 2007) on WS-353, achieving rank correlation of ρ=0.77. This result rivals average human performance, obtaining correlation near that of the supervised oracle approach of Agirre et al. (2009).

This chapter also demonstrates that *feature pruning* is one of the most significant factors in obtaining high correlation with human similarity judgements using vector-space models. Three approaches are evaluated: (1) basic weighted unigram collocations, (2) Explicit Semantic Analysis (ESA; Gabrilovich and Markovitch, 2007), and (3) the *multi-prototype* model. In all three cases we show that feature pruning can be used to significantly improve correlation, in particular reaching the limit of human and oracle performance on WS-353.

- **Selectional Preference** – Predicting the typical filler of an argument slot of a verb (Resnik, 1997; Pantel et al., 2007). In this problem, I show that the tiered clustering model outperforms the other models due to its ability to capture shared structure, particularly in the case of selectionally restrictive verbs (e.g. the set of things that can be *eaten* or can *shoot*).

- **McRae Categorization Norms** – On the McRae category naming task, the multi-prototype model and single prototype models are indistinguishable. However, on the exemplar generation task, the tiered clustering model outperforms the other two.

- **BLESS** – Since it contains multiple similarity relations, the BLESS dataset can help elucidate which axes of similarity different models prefer. The multi-prototype model is biased more towards attributes, events, hypernyms and meronyms, while the single-prototype model and tiered clustering model prefer coordinate terms.

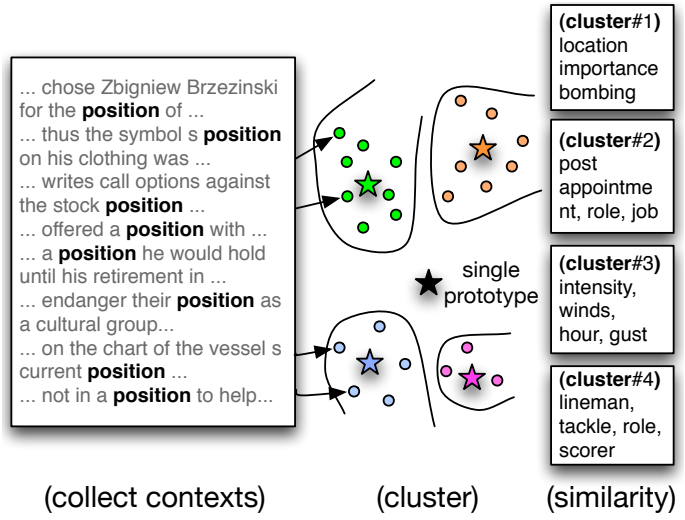


Figure 4.1: Overview of the multi-prototype approach to paraphrase discovery for a single target word independent of context. Occurrences are clustered and cluster centroids are used as prototype vectors. Note the "hurricane" sense of *position* (cluster 3) is not typically considered appropriate in WSD.

4.2 Multi-Prototype Vector-Space Models

The multi-prototype model is similar to standard vector-space models of word meaning, with the addition of a per-word-type clustering step: Occurrences for a specific word type are collected from the corpus and clustered using any appropriate method (§4.2.1). This approach is commonly employed in unsupervised word sense discovery; however, clusters are not intended to correspond to traditional word senses. Rather, clustering is used only to capture meaningful variation in word usage. Similarity between two word types is then computed as a function of their cluster centroids (§4.4.3), instead of the centroid of all the word's occurrences. Figure 4.1 gives an overview of this process.

4.2.1 Spherical Mixture Multi-Prototype Models

Multiple prototypes for each word w are generated by clustering feature vectors $v(c)$ derived from each occurrence $c \in \mathcal{C}(w)$ in a large textual corpus and collecting the resulting cluster centroids $\pi_k(w), k \in [1, K]$. Multiple values of K are evaluated experimentally.

Our experiments employ a *mixture of von Mises-Fisher distributions* (movMF) clustering method with first-order unigram contexts (Banerjee et al., 2005). Feature vectors $v(c)$ are composed of individual features $I(c, f)$, taken as all unigrams $f \in \mathcal{F}$ in a 10-word window around w.

Like spherical k-means (Dhillon and Modha, 2001), movMF models semantic relatedness using cosine similarity, a standard measure of textual similarity. However, movMF introduces an additional per-cluster *concentration* parameter controlling its semantic breadth, allowing it to more accurately model non-uniformities in the distribution of cluster sizes. Based on preliminary experiments comparing various clustering methods, movMF was found to give the best results.

4.2.2 Dirichlet-Process Multi-Prototype Models

One potential issue with the previous model is that K must be chosen and fixed *a priori*. A heuristic solution might be to scale K with the log of the number of word occurrences in the corpus. However, this can be misleading as the total number of occurrences of a word is heavily corpus-dependent, and in particular semantically "tight" corpora such as WSJ high frequency words may have only a small number of senses actually expressed. Furthermore, the number of clusters

should most likely depend on the *variance* of the occurrences, not just the total number.

A more principled, data-driven approach to selecting the number of prototypes per word is to employ a clustering model with infinite capacity, e.g. the Dirichlet Process Mixture Model (DPMM; Neal, 2000; Rasmussen, 2000). The DPMM assigns positive mass to a variable, but finite number of clusters **z**,

$$P(z_i = k|z_{-i}) = \begin{cases} \frac{n_k^{-i}}{\sum_j n_j - 1 + \alpha} & \text{if } n_k > 0 \\ \frac{\alpha}{\sum_j n_j - 1 + \alpha} & k \text{ is a new class.} \end{cases} \tag{4.1}$$

with probability of assignment to cluster k proportional to the number of data points previously assigned to k, n_k. In this case, the number of clusters no longer needs to be fixed a priori, allowing the model to allocate expressivity dynamically to concepts with richer structure. Such a model would allow naturally more polysemous words to adopt more flexible representations.

Instead of assuming all words can be represented by the same number of clusters, representational flexibility can be allocated dynamically using the DPMM. The DPMM is an infinite capacity model capable of assigning data to a variable, but finite number of clusters K_w, with probability of assignment to cluster k proportional to the number of data points previously assigned to k. A single parameter η controls the degree of smoothing, producing more uniform clusterings as $\eta \to \infty$. Using this model, the number of clusters no longer needs to be fixed a priori, allowing the model to allocate expressivity dynamically to concepts with richer structure. Such a model naturally allows the word representation to allocate additional capacity for highly polysemous words, with the number of clusters growing loga-

rithmically with the number of occurrences. The DPMM has been used for rational models of concept organization (Sanborn et al., 2006), but to our knowledge has not yet been applied directly to lexical semantics.

4.3 Tiered Clustering: Multi-Prototype Models with Shared Structure

Tiered clustering implements feature selective clustering by allocating features between two submodels: a (context-dependent) DPMM and a single (context-independent) *background* component. This model is similar structurally to the feature selective clustering model proposed by Law et al. (2002). However, instead of allocating entire feature *dimensions* between model and background components, assignment is done at the level of individual feature occurrences, much like topic assignment in Latent Dirichlet Allocation (LDA; Griffiths et al., 2007b). At a high level, the tiered model can be viewed as a combination of a multi-prototype model and a single-prototype back-off model. However, by leveraging both representations in a joint framework, uninformative features can be removed from the clustering, resulting in more semantically tight clusters.

4.3.1 Generative Model

Concretely, each word occurrence $\mathbf{w}_d$ first selects a cluster $\boldsymbol{\phi}_d$ from the DPMM; then each feature $w_{i,d}$ is generated from either the background model $\boldsymbol{\phi}_{\text{back}}$ or the selected cluster $\boldsymbol{\phi}_d$, determined by the tier indicator $z_{i,d}$. The full generative

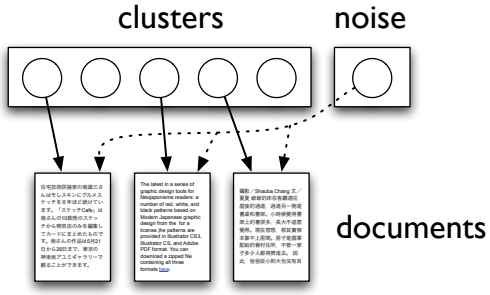


Figure 4.2: Schematic of word occurrences being generated by the tiered clustering model. Each context feature comes from either from the word-dependent cluster component or from the word-independent background component.

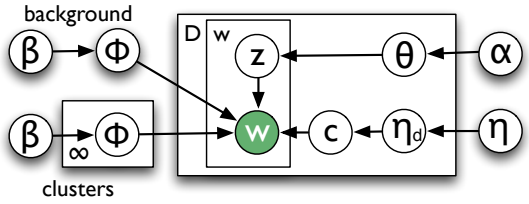


Figure 4.3: Plate diagram for the tiered clustering model with cluster indicators drawn from the Chinese Restaurant Process.

model for tiered clustering is given by

$$
\begin{array}{rcll}
\theta_d|\alpha & \sim & \text{Beta}(\alpha) & d \in D, \\
\boldsymbol{\phi}_d|\boldsymbol{\beta}, G_0 & \sim & \text{DP}(\boldsymbol{\beta}, G_0) & d \in D, \\
\boldsymbol{\phi}_{\text{back}}|\boldsymbol{\beta}_{\text{back}} & \sim & \text{Dirichlet}(\boldsymbol{\beta}_{\text{back}}) & \\
z_{i,d}|\theta_d & \sim & \text{Bernoulli}(\theta_d) & i \in |\mathbf{w}_d|, \\
w_{i,d}|\boldsymbol{\phi}_d, z_{i,d} & \sim & \begin{cases} \text{Mult}(\boldsymbol{\phi}_{\text{back}}) \text{ if } z_{i,d} = 1 \\ \text{Mult}(\boldsymbol{\phi}_d) \text{ otherwise} \end{cases} & i \in |\mathbf{w}_d|,
\end{array}
$$

where $\boldsymbol{\alpha}$ controls the per-data tier distribution smoothing and $\boldsymbol{\beta}$ controls the uniformity of the DP cluster allocation. The DP is parameterized by a base measure G_0, controlling the per-cluster term distribution smoothing; which use a Dirichlet with hyperparameter η, as is common (Figure 4.3).

Since the background topic is shared across all occurrences, it can account for features with *context-independent* variance, such as stop words and other high-frequency noise, as well as the central tendency of the collection (Table 4.2). Furthermore, it is possible to put an asymmetric prior on η, yielding more fine-grained control over the assumed *uniformity* of the occurrence of noisy features, unlike in the model proposed by Law et al. (2002).

4.3.2 Collapsed Gibbs Sampler

Since there is no closed form for the posterior distribution of the Tiered Clustering model, sampling is necessary to perform model inference. By exploiting conjugacy, the latent variables $\boldsymbol{\theta}$, $\boldsymbol{\phi}$ and η_d can be integrated out, yielding an efficient *collapsed Gibbs sampler*. The likelihood of document d is given by

$$
P(\mathbf{w}_d|\mathbf{z}, c_d, \boldsymbol{\phi}) = \prod_i P(w_{i,d}|\boldsymbol{\phi}_{c_d})^{\delta(z_{d,i}=0)} P(w_{i,d}|\boldsymbol{\phi}_{\text{background}})^{\delta(z_{d,i}=1)}. \tag{4.2}
$$

Hence, this model can be viewed as a two-topic variant of LDA with the addition of a per-document cluster indicator.[1] The update rule for the latent tier indicator $\mathbf{z}$ is similar to the update rule for 2-topic LDA, with the background component as the first topic and the second topic being determined by the per-document cluster indicator $\mathbf{c}$.

We can efficiently approximate $p(\mathbf{z}|\mathbf{w})$ via Gibbs sampling, which requires the complete conditional posteriors for all $z_{i,d}$. These are

$$P(z_{i,d} = t|\mathbf{z}_{-(i,d)}, \mathbf{w}, \alpha, \beta) = \frac{n_t^{(w_{i,d})} + \beta}{\sum_w (n_t^{(w)} + \beta)} \frac{n_t^{(d)} + \alpha}{\sum_j (n_j^{(d)} + \alpha)}. \tag{4.3}$$

where $\mathbf{z}_{-(i,d)}$ is shorthand for the set $\mathbf{z} - \{z_{i,d}\}$, $n_t^{(w)}$ is the number of occurrences of word w in topic t not counting $w_{i,d}$ and $n_t^{(d)}$ is the number of words in document d assigned to topic t, not counting $w_{i,d}$.

Likewise sampling the cluster indicators conditioned on the data $p(\mathbf{c}_d|\mathbf{w}, \mathbf{c}_{-d}, \alpha, \eta)$ decomposes into the DP posterior over cluster assignments and the cluster-conditional Multinomial-Dirichlet document likelihood $p(\mathbf{c}_d|\mathbf{w}, \mathbf{c}_{-d}, \alpha, \eta) = p(\mathbf{c}_d|\mathbf{c}_{-d}, \eta)p(\mathbf{w}_d|\mathbf{w}_{-d}, \mathbf{c}, \mathbf{z}, \alpha)$ given by

$$P(c_d = k_{\text{old}}|\mathbf{c}_{-d}, \alpha, \eta) \propto \underbrace{\left(\frac{m_k^{(-d)}}{m_\blacksquare^{(-d)} + \eta}\right)}_{p(\mathbf{c}_d|\mathbf{c}_{-d},\eta)} \underbrace{\frac{C(\alpha + \overrightarrow{\boldsymbol{n}}_k^{(-d)} + \overrightarrow{\boldsymbol{n}}_\blacksquare^{(d)}))}{C(\alpha + \overrightarrow{\boldsymbol{n}}_k^{(-d)})}}_{p(\mathbf{w}_d|\mathbf{w}_{-d},\mathbf{c},\mathbf{z},\alpha)} \tag{4.4}$$

$$P(c_d = k_{\text{new}}|\mathbf{c}_{-d}, \alpha, \eta) \propto \frac{\eta}{m_\blacksquare^{(-d)} + \eta} \frac{C(\alpha + \overrightarrow{\boldsymbol{n}}_\blacksquare^{(d)})}{C(\alpha)} \tag{4.5}$$

[1] Effectively, the tiered clustering model is a special case of the *nested* Chinese Restaurant Process with the tree depth fixed to two (Blei et al., 2003a).

where $m_k^{(-d)}$ is the number of documents assigned to k not including d, $\overrightarrow{\boldsymbol{n}}_k^{(d)}$ is the vector of counts of words from document $\mathbf{w}_d$ assigned to cluster k (i.e. words with $\mathbf{z}_{i,d} = 0$) and $C(\cdot)$ is the normalizing constant for the Dirichlet

$$C(\boldsymbol{a}) = \Gamma(\sum_{j=1}^{m} a_j)^{-1} \prod_{j=1}^{m} \Gamma(a_j)$$

operating over vectors of counts $\boldsymbol{a}$.

4.3.3 Combined Multi-Prototype and Single Prototype

Tiered clustering's ability to model both shared and idiosyncratic structure can be easily approximated by using the single prototype model as the shared component and multi-prototype model as the clustering. Such an *MP+SP* model is conceptually simpler than Tiered clustering and hence easier to implement. However, unlike in the tiered model, all features are assigned to *both* components. This simplification actually hurts performance (§4.5).

4.4 Measuring Semantic Similarity

Computing semantic similarity between multi-prototype and tiered clustering representations is less straightforward than for the single prototype model. This section introduces several simple compound metrics suitable for comparing words with multiple senses.

4.4.1 Multi-prototype Similarity

Due to its richer representational structure, computing similarity in the multi-prototype model is less straightforward than in the single prototype case. One simple approach that performs well and is robust to noise is to average the base similarity scores over all pairs of prototypes (sampled from the cluster distributions) Given two words w and w', this *AvgSim* metric is

$$\text{AvgSim}(w, w') \stackrel{\text{def}}{=} \frac{1}{K_w K_{w'}} \sum_{j=1}^{K_w} \sum_{k=1}^{K_{w'}} d(\pi_k(w), \pi_j(w')) \tag{4.6}$$

K_w and $K_{w'}$ are the number of clusters for w and w' respectively, and $d(\cdot, \cdot)$ is a standard distributional similarity measure (e.g. cosine distance). As cluster sizes become more uniform, AvgSim tends towards the single prototype similarity,[2] hence the effectiveness of AvgSim stems from boosting the influence of small clusters.

As a point of comparison, a second noncontextual similarity metric is

$$\text{MaxSim}(w, w') \stackrel{\text{def}}{=} \max_{1 \le j \le K, 1 \le k \le K} d(\pi_k(w), \pi_j(w'))$$

where $d(\cdot, \cdot)$ is a standard distributional similarity measure. In MaxSim the similarity is the maximum over all pairwise prototype similarities.

In AvgSim, all prototype pairs contribute equally to the similarity computation, thus two words are judged as similar if many of their senses are similar. Note that if all clusters (senses) have equal weight and feature weights are purely additive, then AvgSim devolves into just applying the baseline similarity metric. Since

[2]This can be problematic for certain clustering methods that specify uniform priors over cluster sizes; however the DPMM naturally exhibits a linear decay in cluster sizes with the $\mathbb{E}[\#\text{ clusters of size } M] = \eta/M$.

it weights all senses equally, AvgSim can be seen as a method for increasing the contribution of minority senses to the overall similarity score.

In contrast to AvgSim, MaxSim only requires a single pair of prototypes to be close for the words to be judged similar. Thus, MaxSim models the similarity of words that share only a single sense (e.g. *bat* and *club*) at the cost of lower robustness to noisy clusters that might be introduced when K is large.

A priori one might expect MaxSim to outperform AvgSim, since it compares only the prototypes that are most similar. That is, it would be expected that the tool sense of *bat* would be the most similar prototype to the tool sense of *club*. However, due to e.g. clustering noise and high-frequency head word dependent stopwords, this is not always the case empirically.

4.4.2 Contextual Similarity

When contextual information is available, AvgSim and MaxSim can be modified to produce more precise similarity computations:

$$\begin{aligned}
\text{AvgSimC}(w, w') &\stackrel{\text{def}}{=} \frac{1}{K^2} \sum_{j=1}^{K} \sum_{k=1}^{K} d_{c,w,k} d_{c',w',j} d(\pi_k(w), \pi_j(w')) \\
\text{MaxSimC}(w, w') &\stackrel{\text{def}}{=} d(\hat{\pi}(w), \hat{\pi}(w'))
\end{aligned}$$

where $d_{c,w,k} \stackrel{\text{def}}{=} d(v(c), \pi_k(w))$ is the likelihood of context c belonging to cluster $\pi_k(w)$, and $\hat{\pi}(w) \stackrel{\text{def}}{=} \pi_{\arg\max_{1 \leq k \leq K} d_{c,w,k}}(w)$, the maximum likelihood cluster for w in context c. Thus, AvgSimC corresponds to *soft cluster assignment*, weighting each similarity term in AvgSim by the likelihood of the word contexts appearing in their respective clusters. MaxSimC corresponds to *hard assignment*, using only the

most probable cluster assignment. Note that AvgSim and MaxSim can be thought of as special cases of AvgSimC and MaxSimC with uniform weight to each cluster; hence AvgSimC and MaxSimC can be used to compare words in context to isolated words as well.

4.4.3 Tiered Clustering Similarity

Tiered clustering representations offer more possibilities for computing semantic similarity than multi-prototype, as the background prototype can be treated separately from the other prototypes. I make use of a simple convex combination of the distance between the two background components, and the AvgSim of the two sets of clustering components.

$$\text{TieredAvgSim}(w, w') \stackrel{\text{def}}{=} \alpha AvgSim(w, w') + (1 - \alpha) d(\pi_{\text{back}}(w), \pi_{\text{back}}(w')) \quad (4.7)$$

where $\alpha \in [0, 1]$ controls the tradeoff between the two base similarity measures. In all experiments here we will simply use $\alpha = 0.5$, although tuning α can potentially yield improved results.

4.5 Experimental Results

This section compares four models: (1) the standard single-prototype approach, (2) the multi-prototype approach outlined in §4.2.2, (3) a simple combination of the multi-prototype and single-prototype approaches (MP+SP) and (4) the tiered clustering approach (§4.3.1). Each data set is divided into 5 quantiles based

WordSim-353
stock-live, start-match, line-insurance, game-round, street-place, company-stock
Evocation
break-fire, clear-pass, take-call, break-tin, charge-charge, run-heat, social-play
Padó
see-drop, see-return, hit-stock, raise-bank, see-face, raise-firm, raise-question

Table 4.3: Examples of highly polysemous pairs from each data set using sense counts from WordNet.

homonymous
carrier, crane, cell, company, issue, interest, match, media, nature, party, practice, plant, racket, recess, reservation, rock, space, value
polysemous
cause, chance, journal, market, network, policy, power, production, series, trading, train

Table 4.4: Words used in predicting paraphrases.

on per-pair average sense counts,[3] collected from WordNet 3.0 (Fellbaum, 1998a); examples of pairs in the *high-polysemy* quantile are shown in Table 4.3. Unless otherwise specified, both DPMM multi-prototype and tiered clustering use symmetric Dirichlet hyperparameters, β=0.1, η=0.1, and tiered clustering uses α=10 for the background/clustering allocation smoother.

4.5.1 Predicting Paraphrases

In the following analyses, word occurrences are represented using unordered unigrams collected from a window of size T=10 centered around the occurrence,

[3]Despite many skewed pairs (e.g. line has 36 senses while insurance has 3), I found that arithmetic average and geometric average perform the same.

represented using either *tf-idf* weighting or χ^2 weighting (Agirre et al., 2009; Curran, 2004b). Feature vectors are pruned to a fixed length f, discarding all but the highest-weight features (f is selected via empirical validation, as described in the next section). Finally, the baseline semantic similarity between word pairs is computed using cosine distance (ℓ_2-normalized dot-product).[4]

The multi-prototype model is next evaluated on on its ability to determine the most closely related words for a given target word (using the Wikipedia corpus with *tf-idf* features). The top k most similar words were computed for each prototype of each target word. Using a forced-choice setup, human subjects were asked to evaluate the quality of these *paraphrases* relative to those produced by a single prototype. Participants on Amazon's Mechanical Turk[5] (Snow et al., 2008) were asked to choose between two possible alternatives (one from a prototype model and one from a multi-prototype model) as being most similar to a given target word. The target words were presented either in isolation or in a sentential context randomly selected from the corpus. Table 4.4 lists the ambiguous words used for this task. They are grouped into homonyms (words with very distinct senses) and polysemes (words with related senses). All words were chosen such that their usages occur within the same part of speech.

In the non-contextual task, 79 unique raters completed 7,620 comparisons

[4](**Parameter robustness**) We observe lower correlations on average for T=25 and T=5 and therefore observe T=10 to be near-optimal. Substituting weighted Jaccard similarity for cosine does not significantly affect the results in this chapter.

[5]http://mturk.com

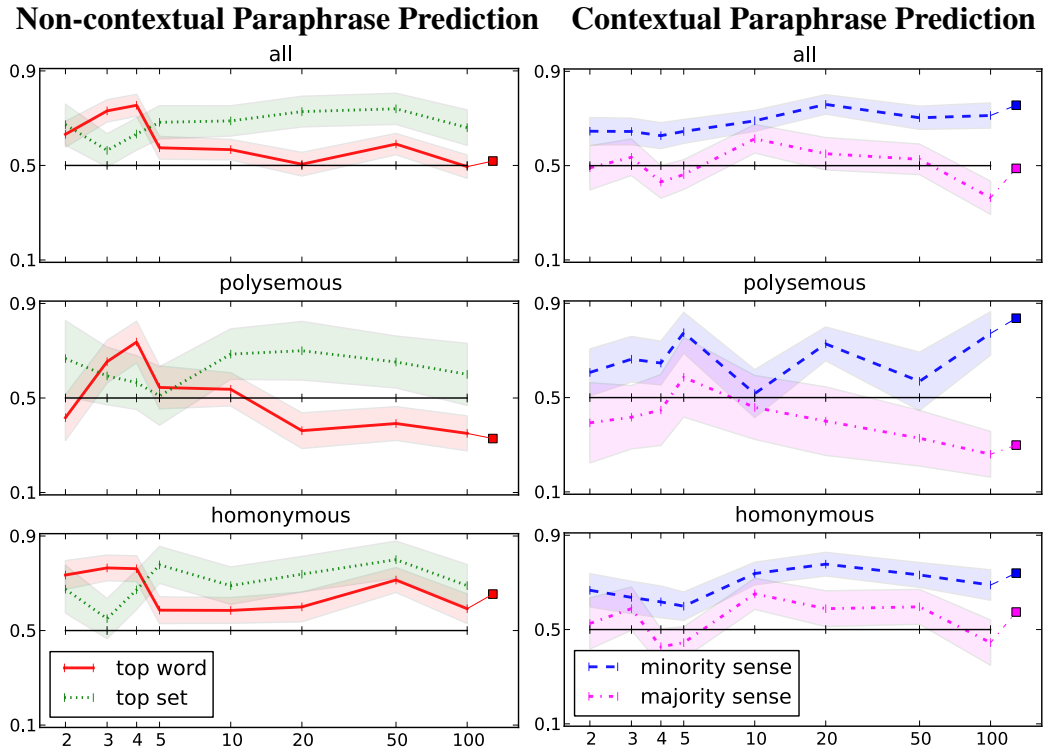


Figure 4.4: (**left**) Paraphrase evaluation for isolated words showing fraction of raters preferring multi-prototype results vs. number of clusters. Colored squares indicate performance when combining across clusterings. 95% confidence intervals computed using the Wald test. (**right**) Paraphrase evaluation for words in a sentential context chosen either from the minority sense or the majority sense.

of which 72 were discarded due to poor performance on a known test set.[6] In the contextual task, 127 raters completed 9,930 comparisons of which 87 were discarded.

[6](**Rater reliability**) The reliability of Mechanical Turk raters is quite variable, so rater quality was evaluated by including control questions with a known correct answers in each HIT. Control questions were generated by selecting a random word from WordNet 3.0 and including as possible choices a word in the same synset (correct answer) and a word in a synset with a high path distance (incorrect answer). Raters who got less than 50% of these control questions correct, or spent too little time on the HIT were discarded.

For the non-contextual case, Figure 4.4 left plots the fraction of raters preferring the multi-prototype prediction (using AvgSim) over that of a single prototype as the number of clusters is varied. When asked to choose between the single best word for each method (**top word**), the multi-prototype prediction is chosen significantly more frequently (i.e. the result is above 0.5) when the number of clusters is small, but the two methods perform similarly for larger numbers of clusters (Wald test, $\alpha = 0.05$.) Clustering more accurately identifies homonyms' clearly distinct senses and produces prototypes that better capture the different uses of these words. As a result, compared to using a single prototype, the multi-prototype approach produces better paraphrases for homonyms compared to polysemes. However, given the right number of clusters, it also produces better results for polysemous words.

The paraphrase prediction task highlights one of the weaknesses of the multi-prototype approach: as the number of clusters increases, the number of occurrences assigned to each cluster decreases, increasing noise and resulting in some poor prototypes that mainly cover outliers. The word similarity task is somewhat robust to this phenomenon, but synonym prediction is more affected since only the top predicted choice is used. When raters are forced to chose between the top *three* predictions for each method (presented as **top set** in Figure 4.4 left), the effect of this noise is reduced and the multi-prototype approach remains dominant even for a large number of clusters. This indicates that although more clusters can capture finer-grained sense distinctions, they also can introduce noise.

When presented with words in context (Figure 4.4 right),[7] raters found no

[7]Results for the multi-prototype method are generated using AvgSimC (soft assignment) as this

significant difference in the two methods for words used in their majority sense.[8] However, when a minority sense is presented (e.g. the "prison" sense of *cell*), raters prefer the choice predicted by the multi-prototype approach. This result is to be expected since the single prototype mainly reflects the majority sense, preventing it from predicting appropriate synonyms for a minority sense. Also, once again, the performance of the multi-prototype approach is better for homonyms than polysemes.

4.5.2 Semantic Similarity: WordSim-353 and Evocation

Figure 4.5 plots Spearman's ρ on WordSim-353 against the number of clusters (K) for Wikipedia and Gigaword corpora, using pruned *tf-idf* and χ^2 features.[9] In general pruned *tf-idf* features yield higher correlation than χ^2 features. Using AvgSim, the multi-prototype approach ($K > 1$) yields higher correlation than the single-prototype approach ($K = 1$) across all corpora and feature types, achieving state-of-the-art results ($\rho = 0.77$) with pruned *tf-idf* features. This result is statistically significant in all cases for *tf-idf* and for $K \in [2, 10]$ on Wikipedia and $K > 4$ on Gigaword for χ^2 features.[10] MaxSim yields similar performance when $K < 10$ but performance degrades as K increases.

---

was found to significantly outperform MaxSimC.

[8] Sense frequency determined using Google; senses labeled manually by trained human evaluators.

[9] (**Feature pruning**) Results using *tf-idf* features are extremely sensitive to feature pruning while χ^2 features are more robust. In all experiments *tf-idf* features are pruned by their overall weight, taking the top 5000. This setting was found to optimize the performance of the single-prototype approach.

[10] Significance is calculated using the large-sample approximation of the Spearman rank *test*; ($p < 0.05$).

Spearman's ρ	**prototype**	**exemplar**	**vMF multi-prototype (AvgSim)**			**MP+SP**
			$K = 5$	$K = 20$	$K = 50$	
Wikipedia *tf-idf*	0.53±0.02	0.60±0.06	0.69±0.02	0.76±0.01	0.76±0.01	0.77±0.01
Wikipedia χ^2	0.54±0.03	0.65±0.07	0.58±0.02	0.56±0.02	0.52±0.03	0.59±0.04
Gigaword *tf-idf*	0.49±0.02	0.48±0.10	0.64±0.02	0.61±0.02	0.61±0.02	0.62±0.02
Gigaword χ^2	0.25±0.03	0.41±0.14	0.32±0.03	0.35±0.03	0.33±0.03	0.34±0.03

Table 4.5: Spearman correlation on the WordSim-353 dataset broken down by corpus and feature type. Results are shown for the vMF multi-prototype model.

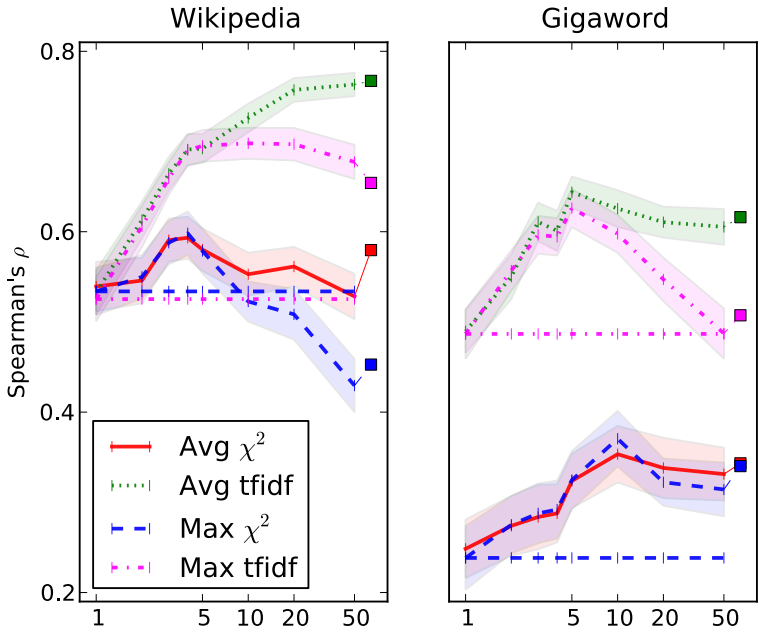


Figure 4.5: WordSim-353 rank correlation vs. number of clusters (log scale) using AvgSim and MaxSim on both the Wikipedia (left) and Gigaword (right) corpora. Horizontal bars show the performance of single-prototype. Squares indicate performance when combining across clusterings. Error bars depict 95% confidence intervals using the Spearman test. Squares indicate performance when combining across clusterings.

It is possible to circumvent the model-selection problem (choosing the best value of K) by simply combining the prototypes from clusterings of different sizes. This approach represents words using both semantically broad and semantically tight prototypes, similar to hierarchical clustering. Table 4.5 and Figure 4.5 (squares) show the result of such a *MP+SP* approach, where the prototypes for clusterings of size 1-5, 10, 20, 50, and 100 are unioned to form a single large prototype set. In general, this approach works about as well as picking the optimal value of K, even outperforming the single best cluster size for Wikipedia.

Finally, the multi-prototype approach is also compared to a pure exemplar approach, averaging similarity across all occurrence pairs.[11] Table 4.5 summarizes the results. The exemplar approach yields significantly higher correlation than the single prototype approach in all cases except Gigaword with *tf-idf* features ($p < 0.05$). Furthermore, it performs significantly *worse* than MP+SP for *tf-idf* features, and does not differ significantly for χ^2 features. Overall this result indicates that multi-prototype performs at least as well as exemplar in the worst case, and significantly outperforms when using the best feature representation / corpus pair.

4.5.2.1 Effects of Pruning

Feature pruning is one of the most significant factors in obtaining high correlation with human similarity judgements using vector-space models, and has been suggested as one way to improve sense disambiguation for polysemous verbs (Xue

[11] Averaging across all pairs was found to yield higher correlation than averaging over the most similar pairs.

Method	WordSim-353			WN-Evocation	
	Sim.	Rel.	Both	Controlled	Turk
Human[a]	0.78	0.74	0.75	0.02	0.37
Agirre et al. (2009)					
best unsup.[b]	0.72	0.56	0.66	-	-
best oracle[c]	**0.83**	0.71	**0.78**	-	-
Single Prototype					
all	0.26	0.29	0.25	0.10	0.10
$f = 1000$	0.76	0.72	0.73	0.21	0.16
$f = 5000$	0.65	0.55	0.59	0.15	0.13
$f = 10000$	0.56	0.46	0.52	0.14	0.12
vMF Multi-Prototype (50 clusters)[d]					
all	0.07	0.17	0.07	0.05	0.08
$f^* = 1000$	0.78	0.70	0.74	**0.25**	**0.16**
$f^* = 5000$	**0.81**	**0.76**	**0.77**	0.24	0.16
$f^* = 10000$	0.79	0.74	0.74	0.24	0.15
Explicit Semantic Analysis					
all	0.58	0.59	0.56	-	-
$f = 1000$	0.75	0.66	0.70	-	-
$f = 5000$	0.77	0.73	0.74	-	-
$f = 10000$	0.77	0.74	0.74	-	-

[a] Surrogate human performance computed using leave-one-out Spearman's ρ averaged across raters for WS-353 and randomized for WN-Evocation. In WN-Evocation, the small number of ratings per pair and randomization makes LOO an unreliable estimator and thus should be interpreted as a rough lower bound.

[b] WordNet-based multilingual approach.

[c] Supervised combination of *b*, context-window features and syntactic features.

[d] Effective number of features, $f^* \stackrel{\text{def}}{=} f/K$ is given in order to enforce a fair comparison.

Table 4.6: Correlation results on WS-353 and WN-Evocation comparing previous studies and surrogate human performance to weighted unigram collocations with feature pruning. Prototype and ESA-based approaches shown use *tf-idf* weighting and cosine distance. Multi-prototype results are given for 50 clusters ($K = 50$).

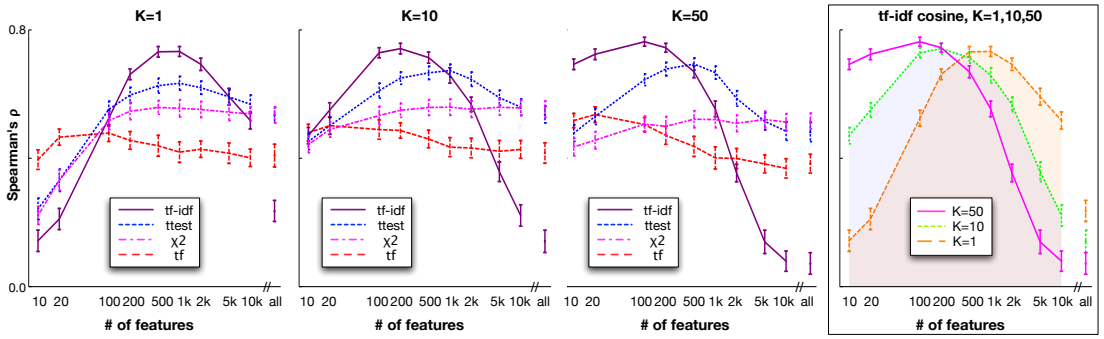


Figure 4.6: Effects of feature pruning and representation on WS-353 correlation broken down across multi-prototype representation size. In general *tf-idf* features are the most sensitive to pruning level, yielding the highest correlation for moderate levels of pruning and significantly lower correlation than other representations without pruning. The optimal amount of pruning varies with the number of prototypes used, with fewer features being optimal for more clusters. Error bars show 95% confidence intervals.

et al., 2006). In this section, the single prototype and multi-prototype methods are calibrated on WS-353, reaching the limit of human and oracle performance and demonstrating robust performance gains even with semantically impoverished features. In particular we obtain ρ=0.75 correlation on WS-353 using *only* unigram collocations and ρ=0.77 using a fixed-K multi-prototype representation (Figure 4.6; Reisinger and Mooney, 2010). This result rivals average human performance, obtaining correlation near that of the supervised oracle approach of Agirre et al. (2009).

In addition to feature weighting, adequate pruning of irrelevant features is critical when computing semantic relatedness. Table 4.6 summarizes the results of using a simple *fixed window* pruning scheme, keeping a fixed number of features (ordered by weight) for each term. Several different feature weighting are evalu-

ated: *tf*, *tf-idf*, *t-test*, and χ^2 (Curran and Moens, 2002). Feature vectors are pruned to a fixed length f, discarding all but the highest-weight features.

For WS-353, unigram collocations perform the worst without pruning (ρ=0.25 for multi-prototype and ρ=0.25 for single prototype), followed by ESA (ρ=0.59), but that with optimal pruning both methods perform about the same (ρ=0.73 and ρ=0.74 respectively). The unpruned multi-prototype approach does poorly with *tf-idf* features because it amplifies feature noise by partitioning the raw occurrences. When employing feature pruning, however, unigram collocations outperform ESA across a wide range of pruning levels. Note that pruning clearly helps in all three test cases and across a wide range of settings for f (cf. Figure 4.6 and Figure 4.7).

For WN-Evocation, there is significant benefit to feature pruning in both the single-prototype and multi-prototype case. The best correlation results are again obtained using pruned *tf-idf* with multiple-prototypes (ρ=0.25 for controlled and ρ=0.16 for Mechanical Turk), although *t-test* features also perform well and benefit from pruning.

The optimal pruning cutoff depends on the feature weighting and number of prototypes (Figure 4.6) as well as the feature representation (Figure 4.7). *t-test* and χ^2 features are most robust to feature noise and perform well even with no pruning; *tf-idf* yields the best results but is sensitive to the pruning parameter. As the number of increases, more pruning is required to combat feature noise.

Figure 4.7 breaks down the similarity pairs into four quantiles for each data set and then shows correlation separately for each quantile. In general the more po-

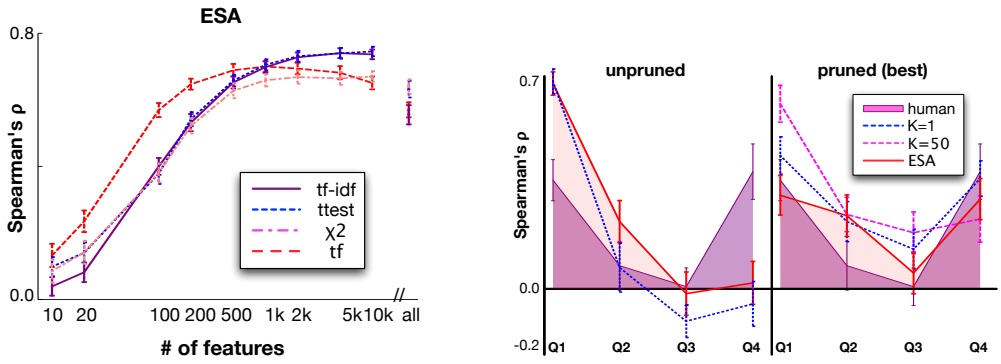


Figure 4.7: (**left**) Effects of feature pruning using ESA on WS-353; more features are required to attain high correlation compared to unigram collocations. (**right**) Correlation results on WS-353 broken down over quantiles in the human ratings. Quantile ranges are shown in Figure 3.1. In general ratings for highly similar (dissimilar) pairs are more predictable (quantiles 1 and 4) than middle similarity pairs (quantiles 2, 3). ESA shows results for a more semantically rich feature set derived using Explicit Semantic Analysis (Gabrilovich and Markovitch, 2007).

larized data quantiles (1 and 4) have higher correlation, indicating that fine-grained distinctions in semantic distance are easier for those sets. The fact that the per-quantile correlation is significantly lower than the full correlation e.g. in the human case means that fine-grained ordering (within quantile) is more difficult than coarse-grained (between quantile). Feature pruning improves correlations in quantiles 2–4 while reducing correlation in quantile 1 (lowest similarity). This result is to be expected as more features are necessary to make fine-grained distinctions between dissimilar pairs.

4.5.2.2 Tiered Clustering and DPMM Multi-Prototype

Correlation results for the tiered clustering model and DPMM multi-prototype model on WS-353 are shown in Table 4.7. In general the approaches incorporating

Method	$\rho \cdot 100$	$\mathbb{E}[C]$	background
Single prototype	73.4±0.5	1.0	-
high polysemy	76.0±0.9	1.0	-
DPMM Multi-prototype	76.8±0.4	14.8	-
high polysemy	79.3±1.3	12.5	-
MP+SP	75.4±0.5	14.8	-
high polysemy	80.1±1.0	12.5	-
Tiered	76.9±0.5	27.2	43.0%
high polysemy	83.1±1.0	24.2	43.0%

Table 4.7: Spearman's correlation on the WS-353 data set. *All* refers to the full set of pairs, *high polysemy* refers to the top 20% of pairs, ranked by sense count. $\mathbb{E}[C]$ is the average number of clusters employed by each method and *background* is the average percentage of features allocated by the tiered model to the background cluster (more features allocated to the background might indicate a higher degree of overlap between senses). 95% confidence intervals are computed via bootstrapping.

multiple prototypes outperform single prototype ($\rho = 0.768$ vs. $\rho = 0.734$). The tiered clustering model does not significantly outperform either the multi-prototype or MP+SP models on the full set, but yields significantly higher correlation on the high-polysemy set.

The tiered model generates more clusters than DPMM multi-prototype (27.2 vs. 14.8), despite using the same hyperparameter settings: Since words commonly shared across clusters have been allocated to the background component, the cluster components have less in common and hence the model splits the data up more finely.

Examples of the tiered clusterings for several words from WS-353 are shown in Table 4.2 and corresponding clusters from the multi-prototype approach are

Method	$\rho \cdot 100$	$\mathbb{E}[C]$	background
Single prototype	19.8±0.6	1.0	-
high similarity	23.9±1.1	1.0	-
high polysemy	11.5±1.2	1.0	-
DPMM Multi-prototype	20.1±0.5	14.8	-
high similarity	22.7±1.2	14.1	-
high polysemy	13.0±1.3	13.2	-
MP+SP	17.6±0.5	14.8	-
high similarity	23.5±1.2	14.1	-
high polysemy	11.4±1.0	13.2	-
Tiered	22.4±0.6	29.7	46.6%
high similarity	27.7±1.3	29.9	47.2%
high polysemy	15.4±1.1	27.4	46.6%

Table 4.8: Spearman's correlation on the Evocation data set. The *high similarity* subset contains the top 20% of pairs sorted by average rater score.

shown in Table 4.1. In general the background component does indeed capture commonalities between all the sense clusters (e.g. all wizards use magic) and hence the tiered clusters are more semantically pure. This effect is most visible in *thematically polysemous* words, e.g. *radio* and *wizard*.

Compared to WS-353, the WN-Evocation pair set is sampled more uniformly from English word pairs and hence contains a significantly larger fraction of unrelated words, reflecting the fact that word similarity is a sparse relation (Figure 3.1 top). Furthermore, it contains proportionally more highly polysemous words relative to WS-353 (Figure 3.1 bottom).

On WN-Evocation, the single prototype and multi-prototype do not differ significantly in terms of correlation (ρ=0.198 and ρ=0.201 respectively; Table 4.8),

Method	$\rho \cdot 100$	$\mathbb{E}[C]$	background
Single prototype	25.8±0.8	1.0	-
high polysemy	17.3±1.7	1.0	-
DPMM Multi-prototype	20.2±1.0	18.5	-
high polysemy	14.1±2.4	17.4	-
MP+SP	19.7±1.0	18.5	-
high polysemy	10.5±2.5	17.4	-
Tiered	29.4±1.0	37.9	41.7%
high polysemy	28.5±2.4	37.4	43.2%

Table 4.9: Spearman's correlation on the Padó data set.

while SP+MP yields significantly lower correlation (ρ=0.176), and the tiered model yields significantly higher correlation (ρ=0.224). Restricting to the top 20% of pairs with highest human similarity judgements yields similar outcomes, with single prototype, multi-prototype and SP+MP statistically indistinguishable (ρ=0.239, ρ=0.227 and ρ=0.235), and tiered clustering yielding significantly higher correlation (ρ=0.277). Likewise tiered clustering achieves the most significant gains on the high polysemy subset.

4.5.3 Selectional Preference

Tiered clustering is a natural model for verb selectional preference, especially for more selectionally restrictive verbs: the set of words that appear in a particular argument slot naturally have some kind of commonality (i.e. they can be *eaten* or can *promise*). The background component of the tiered clustering model can capture such general argument structure. We model each verb argument slot in

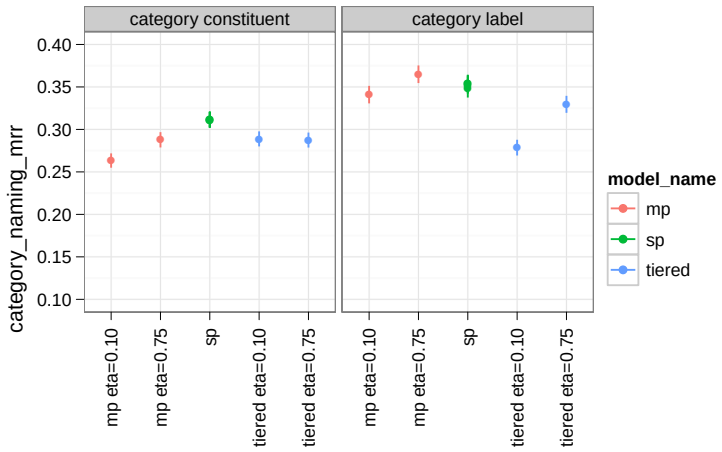
the Padó set with a separate tiered clustering model, separating terms co-occurring with the target verb according to which slot they fill.

On the Padó set, the performance of the DPMM multi-prototype approach breaks down and it yields significantly lower correlation with human norms than the single prototype (ρ=0.202 vs. ρ=0.258; Table 4.9), due to its inability to capture the shared structure among verb arguments. Furthermore combining with the single prototype does not significantly change its performance (ρ=0.197). Moving to the tiered model, however, yields significant improvements in correlation over the other models (ρ=0.294), primarily improving correlation in the case of highly polysemous verbs and arguments.

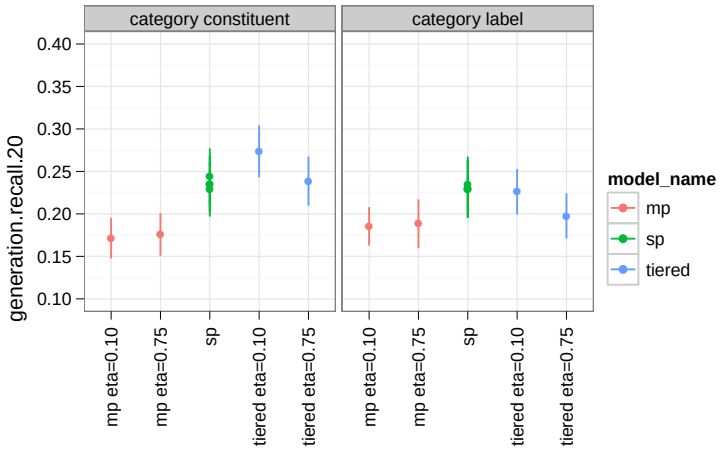
4.5.4 McRae Categorization Norms

In this section and the following sections, five model settings are compared:

1. **sp** – Single prototype model combining all contextual occurrences.
2. **mp** $\eta = 0.75$ – Multi-prototype model with DP concentration $\eta = 0.75$ (fewer clusters).
3. **mp** $\eta = 0.1$ – Multi-prototype model with DP concentration $\eta = 0.1$ (more clusters).
4. **tiered** $\eta = 0.75$ – Tiered clustering model with DP concentration $\eta = 0.75$ (fewer clusters).



(a) **Category Naming** MRR scores on the category-naming task (given an exemplar, predict the category) broken down across the 5 models and 2 category representation types (category-label or category-constituent).



(b) **Exemplar Generation** Recall at rank 20 on the exemplar prediction task (given the category representation, predict individual exemplars) broken down across the 5 models and 2 category representation types.

Figure 4.8

5. **tiered** $\eta = 0.1$ – Tiered clustering model with DP concentration $\eta = 0.1$ (more clusters).

On the McRae dataset, concept-label representations significantly outperform concept-constituent representations on the category naming task, except in the case of tiered clustering with $\eta = 0.1$ (more clusters; Figure 4.8). The best performing models for category naming are the single prototype model and the multi-prototype model with $\eta = 0.75$ (fewer clusters; 0.351 for single prototype vs. 0.365 for multi-prototype). As the number of clusters decreases, the performance of tiered clustering approaches the performance of single prototype. This result indicates that category labels themselves are either homonymous, or else unambiguous; in either case, there is no meaningful shared structure between their constituent senses.

For the exemplar generation task, both the concept-label and concept-constituent representations perform the same, except in the case of tiered clustering where the constituent representation performs significantly better (recall at 20; 0.274 vs. 0.226 for $\eta = 0.1$ and 0.240 vs. 0.197 for $\eta = 0.75$). The best performing model on this task variant is tiered clustering using $\eta = 0.1$ (more clusters; recall of 0.274). The higher performance of tiered clustering on this setup indicates that capturing shared structure between individual exemplar senses is important for capturing how each exemplar relates to the category as whole.

In general, the models capable of capturing background variation (single-prototype and tiered clustering) are better at the exemplar generation task, while

the multi-prototype model is slightly better at category naming.

4.5.5 BLESS

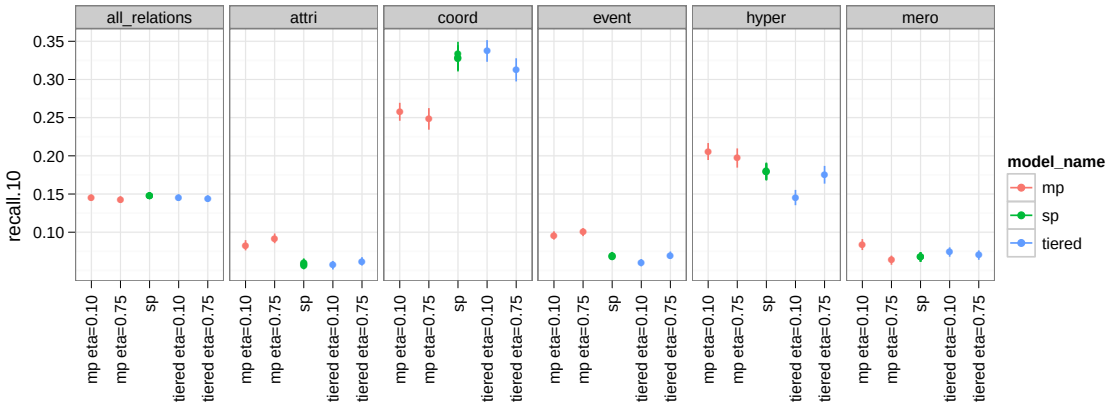
Since it contains multiple axes of similarity relations (attribute, coordinate term, event, hypernym and meronym), the Bless data set is useful for comparing the notion of similarity encoded by each model.

The leftmost panel in Figure 4.9 shows the overall recall across all non-confounder relations (i.e. all relations whose target words were not randomly chosen). Although all three models perform the same in terms of overall recall (recall at 20 of 0.281 for single prototype, 0.271 for multi-prototype, and 0.274 for tiered clustering; Figure 4.9), there are significant differences in what types of related words they prefer.

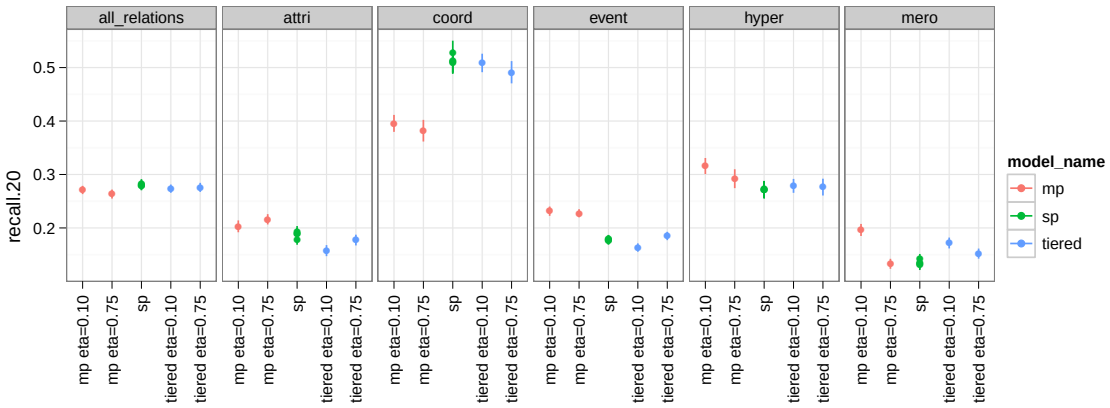
Recall in the multi-prototype model is biased more towards attributes (recall 0.263 for $\eta = 0.75$ vs. 0.187 for single prototype), events (recall 0.228 for $\eta = 0.75$ vs. 0.178 for single prototype), hypernyms (recall 0.316 for $\eta = 0.1$ vs. 0.271 for single prototype) and meronyms (recall 0.197 for $\eta = 0.1$ vs. 0.134 for single prototype) than either single prototype or tiered clustering.

The single prototype and tiered clustering model overwhelmingly prefer coordinate terms (coordinate recall of 0.516 for single prototype and 0.510 for tiered clustering with $\eta = 0.1$).

The multi-prototype model with more clusters ($\eta = 0.1$) prefers hypernyms (recall 0.316) and meronyms (recall 0.196) more and attributes (recall 0.203) sig-



(a) **Recall at 10**



(b) **Recall at 20**

Figure 4.9: Recall scores for each model broken down by relation type. **all_relations** denotes the set of recalled items that were not random confounders.

nificantly less than the same model with fewer clusters ($\eta = 0.75$; hypernym recall 0.292, meronym recall 0.132 and attribute recall 0.216).

These results indicate that coordinate terms are more likely to share similar cooccurrence features than other relation types. Hence, when employing either the single prototype or tiered clustering model, which pool across such features, coordinate recall will be high. In contrast, when used with the AvgSim metric, the multi-prototype model allows for less common features to expressed more strongly, biasing recall towards attributes, events, hypernyms and meronyms.

4.6 Discussion

This chapter introduced two resource-light models for vector-space word meaning that represents words as structured collections of prototype vectors, naturally accounting for lexical ambiguity.

The *multi-prototype* approach uses word sense discovery to partition a word's contexts and construct "sense specific" prototypes for each cluster. Doing so significantly increases the accuracy of lexical-similarity computation as demonstrated by improved correlation with human similarity judgements and generation of better paraphrases according to human evaluators. Furthermore, although performance is sensitive to the number of prototypes, combining prototypes across a large range of clusterings performs nearly as well as the ex-post best clustering.

Compared to WordNet, the best-performing clusterings are significantly more fine-grained. Furthermore, they often do not correspond to agreed upon se-

mantic distinctions (e.g., the "hurricane" sense of *position* in Fig. 4.1). The finer-grained senses are posited to actually capture useful aspects of word meaning, leading to better correlation with WordSim-353.

Feature pruning can significantly improve correlation with human similarity and relatedness judgements. Feature selection combined with the multi-prototype representation achieves state-of-the-art results on the WordSim-353 task, beating a measure of human performance, and performing nearly as well as a supervised oracle approach. The complexity of the interaction between feature weighting and pruning and magnitude of their combined effect on correlation strongly suggests that they should be studied in greater detail, and form a major component of the future work in this thesis.

The multi-prototype model does not lead to improved performance over the single prototype baseline on either McRae category naming or exemplar generation. However, on the BLESS dataset, it yields significantly higher recall for attributes, events, hypernyms and meronyms at the expense of recall on coordinate terms.

The *tiered clustering* model extends the multi-prototype model with additional structure for capturing shared (context-independent) variation in word occurrence features. The ability to model background variation, or shared structure, is shown to be beneficial for modeling words with high polysemy, yielding increased correlation with human similarity judgements modeling word relatedness and selectional preference. Furthermore, the tiered clustering model is shown to significantly outperform related models, yielding qualitatively more precise clusters.

The benefits of this tiered model are most pronounced on a selectional preference task, where there is significant shared structure imposed by conditioning on the verb. Although the results on the Padó are not state of the art,[12] I believe this to be due to the impoverished vector-space design; tiered clustering can be applied to more expressive vector spaces, such as those incorporating dependency parse and FrameNet features.

The tiered clustering model outperforms both other models on the exemplar generation task on the McRae dataset, suggesting that it is a better model of the exemplar structure of human concepts. However, on the BLESS data, tiered clustering performance is similar to the single-prototype model, implying that the existence of the background component has a significant effect on the types of relations captured.

One potential explanation for the superior performance of the tiered model vs. the DPMM multi-prototype model is simply that it allocates more clusters to represent each word (Reisinger and Mooney, 2010). However, decreasing the hyperparameter β (decreasing vocabulary smoothing and hence increasing the effective number of clusters) beyond $\beta = 0.1$ actually hurts multi-prototype performance. The additional clusters do not provide more semantic content due to significant background similarity.

[12] E.g., Padó et al. (2007) report ρ=0.515 on the same data.

Chapter 5

Cross-Cutting Models of Lexical Semantics

5.1 Introduction

Humans categorize objects using multiple orthogonal taxonomic systems, where category generalization depends critically on what features are relevant to one particular system. For example, foods can be organized in terms of their nutritional value (high in fiber) or situationally (commonly eaten for Thanksgiving; Shafto et al. (2006)). Human knowledge-bases such as Wikipedia also exhibit such *cross-cutting* taxonomic structure (e.g. people are organized by occupation or by nationality).

The existence of cross-cutting structure can be explained by multiple competing subsets of salient features: As feature dimensionality increases, the number of ways the data can exhibit interesting structure goes up exponentially. Models such as *Cross-Cutting Categorization* (Cross-cat; (Mansinghka et al., 2006, 2009)) account for this structure by assigning concept features to one of several views, and then clustering the data separately with each view. This approach yields multiple orthogonal clusterings and isolates the effects of noisy features.

This thesis posits that, since the effects of overlapping categorization systems are apparent at the lexical semantic level (Murphy, 2002) as well, lexico-

graphical word senses and traditional computational models of word-sense based on clustering or exemplar activation are potentially too impoverished to capture the rich dynamics of word usage. Different subsets of features may yield different sense views; e.g. clustering using only syntactic features vs. clustering using only document co-occurrence features.

In lexical semantics, context-dependent word similarity can be computed over multiple cross-cutting dimensions. For example, *lung* and *breath* are similar thematically, while *authoritative* and *superficial* occur in similar syntactic contexts, but share little semantic similarity. Both of these notions of similarity play a role in determining word meaning, and hence lexical semantic models should ideally take them both into account.

This chapter introduces a set of novel probabilistic lexical semantics models based on Latent Dirichlet Allocation (LDA) that find multiple overlapping feature subsets, corresponding to principle axes of variation in concepts (Griffiths et al., 2007b). Such *Multi-view* models (MVM) are flexible enough to capture both variation due to syntactic context features as well as higher level thematic features, e.g., be used to capture both *syntagmatic* and *paradigmatic* notions of word meaning. The end result is a model capable of representing multiple, overlapping similarity metrics that result in disparate valid clusterings leveraging the

> **Subspace Hypothesis**: For any pair of words, the set of "active" features governing their apparent similarity differs. For example *wine* and *bottle* are similar and *wine* and *vinegar* are similar, but it would not be

reasonable to expect that the features governing such similarity computations to overlap much, despite occurring in similar documents.

MVM can extract multiple competing notions of similarity, for example both *paradigmatic*, or thematic similarity, and *syntagmatic* or syntactic similarity, in addition to more fine grained relations.

In this chapter I introduce three Multi-View models that encode similarity across multiple overlapping dimensions, and demonstrate ways in which such models can capture context-dependent variation in word usage can be accounted:

1. **Multi-View Assignment** (MV-A) – Words features are distributed across multiple *views* using LDA capturing broad patterns in their syntactic or semantic usage (§5.2.1).

2. **Multi-View Clustering** (MV-C) – Words are assigned to multiple clusterings (views) based on different subsets of features, subject to the marginal constraint that feature subsets are distributed according to LDA (§5.2.2). MV-C combines primitives from Dirichlet-Process Mixture Models (DPMMs) and LDA. Each clustering in MV-C consists of a distribution over features and data and views are further subdivided into clusters based on a DPMM. Hence, each view produces a clustering based on a weighted subset of the available features, allowing more flexibility and robustness than traditional clustering methods.

3. **Multi-View Vector Space** (MV-VS) – Each view is used to contribute a copy of each word feature with weight determined by the underlying Multi-View

model. Hence, the model is responsible for producing multiple *senses* of each feature, increasing the representational power of the vector space (§5.2.3).

The three MVM models are evaluated both

1. According to the human-interpretability of their internal structure; directly measuring their purity as clustering methods (§5.3), and

2. As word representations in the battery of common lexical semantic tasks introduced in §3.2 (§5.4).

In the human-interpretability studies, MV-C is shown to find more semantically and syntactically coherent fine-grained structure, using both common and rare n-gram contexts. Furthermore, MV-A is shown to yield better recall for events, hypernyms, meronyms and attributes, while MV-VS is shown to yield significant improvements over the baseline vector-space model in lexical substitution.

5.2 Multi-View Lexical Semantic Models

This section introduces the basic Latent Dirichlet Allocation structure underlying Multi-view models (MV-A) and then derives two additional sub-models: (1) the Multi-view Clustering Model (MV-C), where words are clustered within each view, and (2) Multi-view Vector Space Model (MV-VS), where views are used to augment the number of features in the model.

5.2.1 Multi-View Model

The basis for Multi-view lexical semantics models is *Latent Dirichlet Allocation* (LDA) a fully Bayesian extension of LSA (Deerwester et al., 1990). In LDA, a set of data $\mathcal{D} = \{\mathbf{w}_d | d \in [1 \dots D]\}$ is projected onto $|M|$ disparate views, which capture the major axes of variation in the features.

Each data vector $\mathbf{w}_d$ consists of context features and associated frequencies collected for word d, for example unigram co-occurrences 3.1.2 or Wikipedia document names. Data vectors are associated with a probability distribution over views $\boldsymbol{\theta}_d^{|M|}$. Empirically, $\boldsymbol{\theta}_d^{|M|}$ is represented as a set of *feature-view* assignments $\mathbf{z}_d$, sampled via the standard LDA collapsed Gibbs sampler.

Each view maintains a separate distribution over features. The generative model for feature-view assignment is given by

$$\begin{array}{rcll} \boldsymbol{\theta}_d^{|M|} | \boldsymbol{\alpha} & \sim & \text{Dirichlet}(\boldsymbol{\alpha}), & d \in D, \\ \boldsymbol{\phi}_m | \boldsymbol{\beta} & \sim & \text{Dirichlet}(\boldsymbol{\beta}), & m \in |M|, \\ z_{dn} | \boldsymbol{\theta}_d & \sim & \text{Discrete}(\boldsymbol{\theta}_d), & n \in |\mathbf{w}_d|, \\ w_{dn} | \boldsymbol{\phi}_{z_{dn} m} & \sim & \text{Discrete}(\boldsymbol{\phi}_{z_{dn} m}), & n \in |\mathbf{w}_d|, \end{array}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are hyperparameters smoothing the per-document topic distributions and per-topic word distributions respectively.

Intuitively, each $\boldsymbol{\phi}$ vector encodes a coherent feature view, or principle component of the original feature space. Likewise, since the cardinality of the latent $\mathbf{z}$ assignments is significantly smaller than the original feature vocabulary, they induce a soft clustering of the features. Dinu and Lapata (2010) term the $\mathbf{z}$ assignments the latent *senses* of the features. The sections will build on this notion, intro-

ducing a multiple clustering procedure (MV-C) and vector space model (MV-VA) that makes explicit use of these latent sense groupings.

5.2.1.1 Contextual Representation

In order to incorporate contextual information into the vector space, the word-conditional probabilities $p(z|w)$ can be replaced with $p(\mathbf{z}|w, \{c_w\})$

$$\mathbf{v}(\mathbf{w}|\{c_w\}) = (p(z_1|\mathbf{w}, \{c_w\}), \ldots, p(z_M|\mathbf{w}, \{c_w\})). \tag{5.1}$$

Two derivations of $p(z_m|\mathbf{w}, \{c_w\})$ are considered in this thesis:

- **DL10**: Dinu and Lapata (2010)

  The probabilities $p(z_m|\mathbf{w}, \{c_w\})$ for $m \in [1..M]$ can be factorized into the product of the join probability of the target word $\mathbf{w}$ and each view z_m, $p(\mathbf{w}, z_m)$ and the conditional probability of context set $\{c_w\}$ given $\mathbf{w}$ and z_m, $p(\{c_w\}|z_m, \mathbf{w})$:

  $$p(z_m|\mathbf{w}, \{c_w\}) = \frac{p(\mathbf{w}, z_m)p(\{c_w\}|z_m, \mathbf{w})}{\sum_k p(\mathbf{w}, z_k)p(\{c_w\}|z_k, \mathbf{w})}. \tag{5.2}$$

  Since $p(\{c_w\}|z_m, \mathbf{w})$ is high-dimensional and hence difficult to estimate, following Dinu and Lapata (2010), by making the simplifying assumption that the target words $\mathbf{w}$ and context words $\{c_w\}$ are conditionally independent given the view z_m, i.e. $p(\{c_w\}|z_m, \mathbf{w})$ can be approximated by $p(\{c_w\}|z_m)$:

  $$p(z_m|\mathbf{w}, \{c_w\}) \approx \frac{p(\mathbf{w}|z_m)p(\{c_w\}|z_m)}{\sum_k p(\mathbf{w}|z_k)p(\{c_w\}|z_k)} \tag{5.3}$$

Equation 5.3 embedded in the vector space from equation 5.1 forms the basis of the **DL10** model-based contextualization method.

- **OK11**: Ó Séaghdha and Korhonen (2011)

  Starting with modeling $p(\{c_w\}|\mathbf{w})$, the conditional probability of the context set $\{c_w\}$ given the target word $\mathbf{w}$ can be broken out over the latent views to yield a vector space model similar to **DL10**:

$$p(\{c_w\}|\mathbf{w}) = \sum_k p(\{c_w\}|z_k)p(z_k|\mathbf{w}) \tag{5.4}$$

$$p(z|\mathbf{w},\{c_w\}) = \frac{p(z|\mathbf{w})p(\{c_w\}|\mathbf{w})}{\sum_k p(z_k|\mathbf{w})p(\{c_w\}|\mathbf{w})} \tag{5.5}$$

  where $p(\{c_w\}|z_k) = \prod_{cin\{c_w\}} p(c|z_k)$ (making use of the assumption that context words are independent given the view assignments). Equation 5.5 can be viewed as a product of experts model (Hinton, 2002), with each additional context word contributing multiplicatively to the conditional likelihood.

The main difference in **DL10** and **OK11** is that the latter makes use of the marginal probability of the context set $\{c_w\}$ given the target word $\mathbf{w}$, $p(\{c_w\}|\mathbf{w})$, while the former makes the simplifying assumption that this can be approximated by $p(\{c_w\}|z_m)$. In practice the difference between these two contextualization strategies are not statistically significant, so all results presented will be using **OK11**.

5.2.2 Multi-View Clustering

MV-A embeds all words in a single metric space and hence posits a globally consistent metric that captures word similarity. Rather than assuming such a global

metric embedding exists, MV-C leverages the *cluster assumption*, e.g. that similar words should appear in the same clusters, in particular extending it to multiple clusterings. The cluster assumption is a natural fit for lexical semantics, as partitions can account for metric violations.

Clustering is commonly used to explain data, but often there are several equally valid, competing clusterings, keying off of different subsets of features, especially in high-dimensional settings such as text mining (Niu et al., 2010). For example, company websites can be clustered by sector or by geographic location, with one particular clustering becoming predominant when a majority of features correlate with it. In fact, informative features in one clustering may be noise in another, e.g. the occurrence of *CEO* is not necessarily discriminative when clustering companies by industry sector, but may be useful in other clusterings. Multiple clustering is one approach to inferring feature subspaces that lead to high quality data partitions. Multiple clustering also improves the flexibility of generative clustering models, as a single model is no longer required to explain all the variance in the feature dimensions (Mansinghka et al., 2009).

MV-C is a multinomial-Dirichlet multiple clustering procedure for distributional lexical semantics that fits multiple, overlapping Dirichlet Process Mixture Models (DPMM) to a set of word data. Features are distributed across the set of clusterings (views) using LDA (as in MV-A), and each DPMM is fit using a subset of the features. This reduces clustering noise and allows MV-C to capture multiple ways in which the data can be partitioned. Figure 5.1 shows a simple example, and Figure 5.2 shows a larger sample of feature-view assignments from a 3-view MV-C

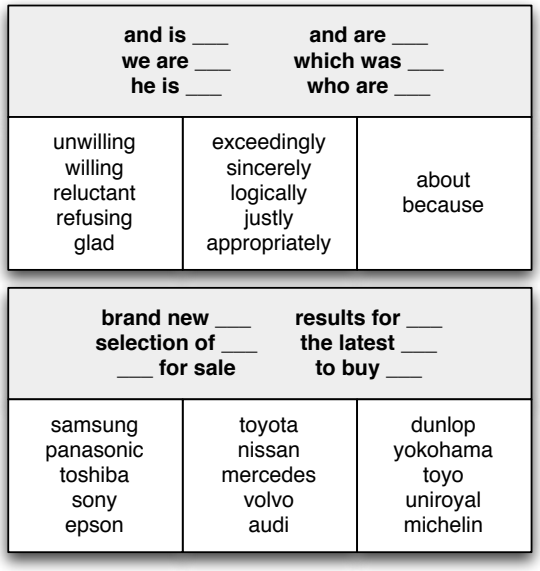


Figure 5.1: Example clusterings from MV-C applied to Google n-gram data. Top contexts (features) for each view are shown, along with examples of word clusters. The top view contains syntactic features that yield personal attributes (e.g. adjectives and adverbs), while the bottom view contains patterns for online consumer goods. Although these particular examples are interpretable, in general the relationship captured by the view's context subspace is not easily summarized.

fit to contexts drawn from the Google n-gram corpus.

MV-C can be implemented using generative model primitives drawn from Latent Dirichlet Allocation (LDA) and the Dirichlet Process (DP). Conditional on the feature-view assignment $\{\mathbf{z}\}$, a clustering is inferred for each view using the Chinese Restaurant Process representation of the DP. The clustering probability is given by

$$\begin{aligned} p(\mathbf{c}|\mathbf{z},\mathbf{w}) &\propto p(\{\mathbf{c}_m\},\mathbf{z},\mathbf{w}) \\ &= \prod_{m=1}^{M}\prod_{d=1}^{|D|} p(\mathbf{w}_d^{[\mathbf{z}=m]}|\mathbf{c}_m,\mathbf{z})p(\mathbf{c}_m|\mathbf{z}). \end{aligned}$$

where $p(\mathbf{c}_m|\mathbf{z})$ is a prior on the clustering for view m, i.e. the DPMM, and $p(\mathbf{w}_d^{[\mathbf{z}=m]}|\mathbf{c}_m,\mathbf{z})$ is the likelihood of the clustering $\mathbf{c}_m$ given the data point $\mathbf{w}_d$ restricted to the features assigned to view m:

$$\mathbf{w}_d^{[\mathbf{z}=m]} \stackrel{\text{def}}{=} \{w_{id}|z_{id}=m\}.$$

Thus, the m clusterings $\mathbf{c}_m$ are treated as conditionally independent given the feature-view assignments.

The feature-view assignments $\{\mathbf{z}\}$ act as a set of marginal constraints on the multiple clusterings, and the impact that each data point can have on each clustering is limited by the number of features assigned to it. For example, in a two-view model, $z_{id}=1$ might be set for all syntactic features (yielding a syntagmatic clustering) while $z_{id}=2$ is set for document features (paradigmatic clustering).

By allowing the clustering model capacity to vary via the DPMM, MV-C can naturally account for the semantic variance of the view. This provides a novel

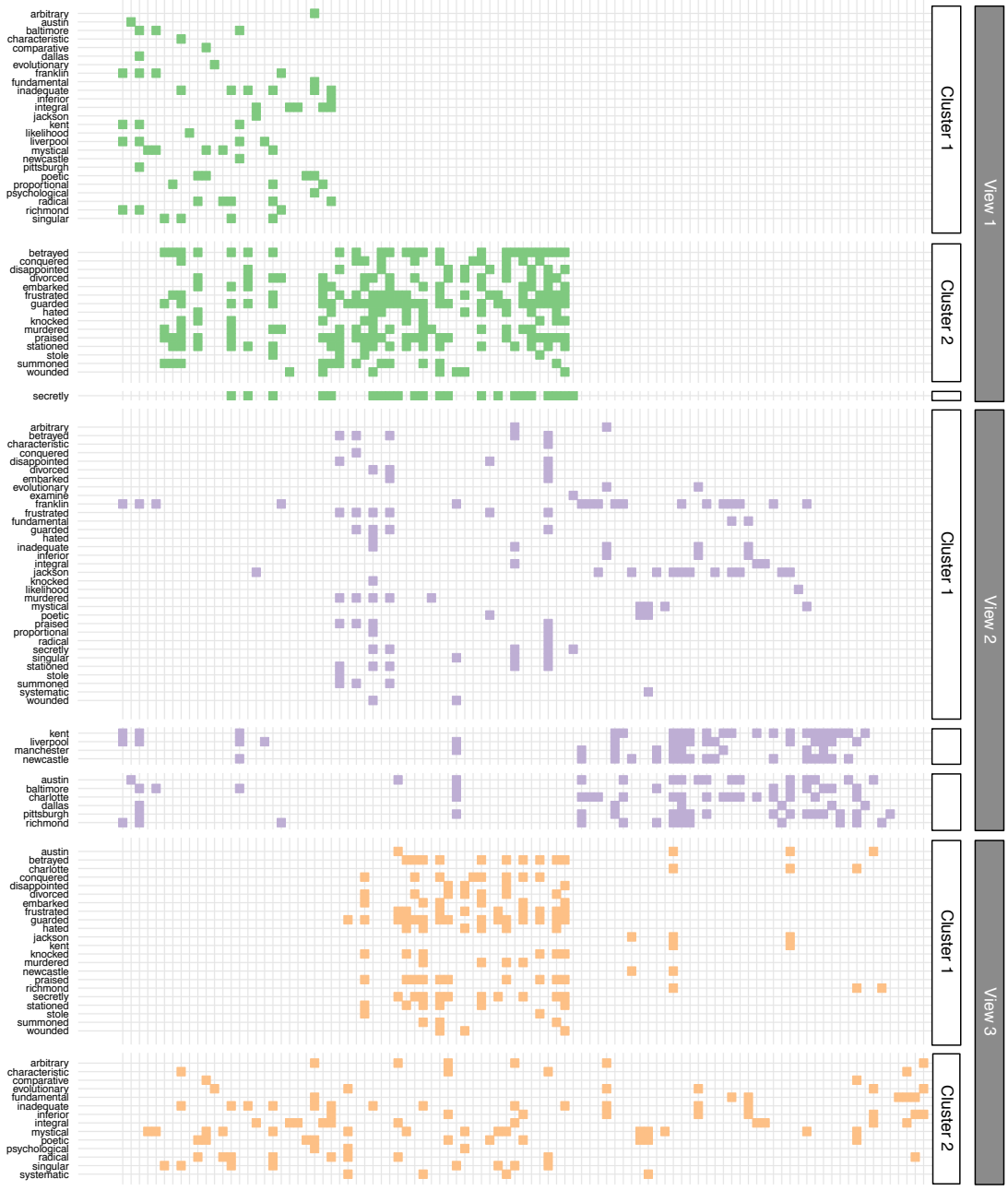


Figure 5.2: (Caption opposite page)

Figure 5.2: **Topics with Senses**: A 3-view MV-C model fit to the Google n-gram context data. Columns show the top 20% of features across all views, while rows show individual data points (words) divided by cluster and view.

Different views place different mass on different sets of features. For example, view 1 puts most of its mass on the first half of the syntactic features shown, while view 3 spreads its mass out over move features.

Words are clustered based on these overlapping subsets. For example, view 1 cluster 2 and View 3 cluster 1 both contain past-tense verbs, but only overlap on a subset of syntactic features.

mechanism for handling feature noise: noisy features can be assigned to a separate view with potentially a small number of clusters. This phenomenon is apparent in cluster 1, view 1 in the example in figure 5.2, where place names and adjectives are clustered together using rare contexts

From a topic modeling perspective, MV-C finds topic refinements within each view, similar to hierarchical methods such as the nested Chinese Restaurant Process (Blei et al., 2003a). The main difference is that the features assigned to the second "refined topics" level are constrained by the higher level, similar to hierarchical clustering. Unlike hierarchical clustering, however, the top level topics/views form an admixture, allowing individual features from a single data point to be assigned to multiple views.

The most similar model to MV-C is *Cross-cutting categorization*, which fits multiple DPMMs to non-overlapping partitions of features (Mansinghka et al., 2009; Shafto et al., 2006). Unlike MV-C, Cross-cat *partitions* features among multiple DPMMs, hence all occurrences of a particular feature will end up in a single

clustering, instead of assigning them softly using LDA. Such hard feature partitioning does not admit an efficient sampling procedure, and hence Shafto et al. (2006) rely on Metropolis-Hastings steps to perform feature assignment, making the model less scalable.

MV-C is also similar to the multiple disparate clusterings framework proposed by Jain et al. (2008). In that work, all clusterings use all features, and hence robustness to feature noise is not treated. MV-C is more similar to the model proposed by Cui et al. (2007), which generates a maximally orthogonal cluster ensemble (cf. Azimi and Fern, 2009; Strehl and Ghosh, 2003). The data are repeatedly projected onto the space most orthogonal to the current clustering and then reclustered.

Given such word representation data, MV-C generates a fixed set of M context views corresponding to dominant eigenvectors in local syntactic or semantic space. Within each view, MV-C partitions words into clusters based on each word's *local representation* in that view; that is, based on the set of context features it allocates to the view. Words have a non-uniform affinity for each view, and hence may not be present in every clustering (Figure 5.2). This is important as different ways of drawing distinctions between words do not necessarily apply to all words. In contrast, LDA finds locally consistent collections of contexts but does not further subdivide words into clusters given that set of contexts. Hence, it may miss more fine-grained structure, even with increased model complexity.

5.2.2.1 Inference

In order to approximate MV-C, we derive a two-stage sampling process, enforcing independence between the LDA component and the clustering components. Although this assumption violates the generative semantics of the model and potentially leads to inconsistent conditional distributions, we find that in practice this does not adversely affect clustering quality. Relaxing these assumptions is one active area of future work.

Inference by collapsed Gibbs sampling proceeds in rounds, alternatingly sampling from $p(\mathbf{z}|\mathbf{w})$ and $p(\mathbf{c}|\mathbf{w}, \mathbf{z})$. $p(\mathbf{z}|\mathbf{w})$ is approximated using the standard LDA collapsed Gibbs sampler, exploiting Multinomial-Dirichlet conjugacy, marginalizing out $\mathbf{c}$:

$$P(z_{i,d} = m|\mathbf{z}_{-(i,d)}, \mathbf{w}, \alpha, \beta) \quad = \quad \frac{n_m^{(w_{i,d})} + \beta}{\sum_w (n_m^{(w)} + \beta)} \frac{n_m^{(d)} + \alpha}{\sum_j (n_j^{(d)} + \alpha)}$$

where $\mathbf{z}_{-(i,d)}$ is shorthand for the set $\mathbf{z} - \{z_{i,d}\}$, $n_m^{(w)}$ is the number of occurrences of word w in view t not counting $w_{i,d}$ and $n_m^{(d)}$ is the number of features in occurrence d assigned to view m, not counting $w_{i,d}$.

Conditional on the feature-view assignments $\mathbf{z}$, word-cluster assignments via the Chinese Restaurant Process view of the DP; $p(\mathbf{c}_d|\mathbf{w}, \mathbf{c}_{-d}, \alpha, \eta)$ decomposes into the DP posterior over cluster assignments and the cluster-conditional Multinomial-Dirichlet word-occurrence likelihood $p(\mathbf{c}_d|\mathbf{w}, \mathbf{c}_{-d}, \alpha, \eta) = p(\mathbf{c}_d|\mathbf{c}_{-d}, \eta)p(\mathbf{w}_d|\mathbf{w}_{-d}, \mathbf{c}, \mathbf{z}, \alpha)$

given by

$$P(c_d = k_{\text{old}}|\mathbf{c}_{-d}, \alpha, \eta) \propto \underbrace{\left(\frac{m_k^{(-d)}}{m_{\blacksquare}^{(-d)} + \eta}\right)}_{p(\mathbf{c}_d|\mathbf{c}_{-d},\eta)} \underbrace{\frac{C(\alpha + \overrightarrow{\boldsymbol{n}}_k^{(-d)} + \overrightarrow{\boldsymbol{n}}_{\blacksquare}^{(d)}))}{C(\alpha + \overrightarrow{\boldsymbol{n}}_k^{(-d)})}}_{p(\mathbf{w}_d|\mathbf{w}_{-d},\mathbf{c},\mathbf{z},\alpha)}$$

$$P(c_d = k_{\text{new}}|\mathbf{c}_{-d}, \alpha, \eta) \propto \frac{\eta}{m_{\blacksquare}^{(-d)} + \eta} \frac{C(\alpha + \overrightarrow{\boldsymbol{n}}_{\blacksquare}^{(d)})}{C(\alpha)}$$

where $m_k^{(-d)}$ is the number of occurrences assigned to k not including d, $\overrightarrow{\boldsymbol{n}}_k^{(d)}$ is the vector of counts of words from occurrence $\mathbf{w}_d$ assigned to cluster k (i.e. words with $\mathbf{z}_{i,d} = 0$) and $C(\cdot)$ is the normalizing constant for the Dirichlet $C(\boldsymbol{a}) = \Gamma(\sum_{j=1}^m a_j)^{-1} \prod_{j=1}^m \Gamma(a_j)$ operating over vectors of counts $\boldsymbol{a}$.

5.2.2.2 Vector Space Representations

Following Dinu and Lapata (2010) and Ó Séaghdha and Korhonen (2011), the word-conditional view assignment probabilities can be cast into a vector space

$$\mathbf{v}(\mathbf{w}) = (p(z_1|\mathbf{w}), \ldots, p(z_M|\mathbf{w})) \tag{5.6}$$

i.e. the vector of LDA topic proportions conditioned on w. Since the total conditional probabilities sum to one, *Bhattacharyya distance* is a natural choice of metric on this space:

$$d_{\text{BC}}(\mathbf{w}, \mathbf{w}') \stackrel{\text{def}}{=} \sqrt{\mathbf{v}(\mathbf{w})}^\top \sqrt{\mathbf{v}(\mathbf{w}')} = \sum_z \sqrt{p(z|\mathbf{w})p(z|\mathbf{w}')} \tag{5.7}$$

Bhattacharyya distance is an approximation of the relative similarity of two distributions and is closely related to the Hellinger distance (Bhattacharyya, 1943).

For typical settings of M, this representation is significantly more compact than the underlying raw vector space, and hence it can be considered a form of lossy

compression. Indeed, as demonstrated in §5.4, this representation performs poorly compared to the raw vector space, although similarity computation is significantly faster given a trained model. The MV-VS model described in §5.2.3 addresses this issue.

5.2.3 Multi-View Vector Space Model

This section introduces the *multi-view vector space model* (MV-VS), combining the fine-grained features representation of first-order vector space models with MVM's ability to identify multiple feature subspaces.

5.2.3.1 Basic Model

In MV-VS, word data is fit using MV-A (§5.2.1), resulting in latent view assignments z_{id} for each feature $w_{id} \in \mathbf{w}_d$. An *augmented* vector representation of $\mathbf{w}_d$ is constructed by duplicating each word feature once per view weighted by the probability of being generated by each view $p(w_{di}|z_m, \mathbf{w}_d)$. That is

$$\mathbf{v}_{\text{MV-VS}}(\mathbf{w}_d) = (f(w_{di})p(z_1|\mathbf{w}), \ldots, f(w_{di})p(z_M|\mathbf{w})|w_{di} \in \mathbf{w}_d) \tag{5.8}$$

where $f(w_{di})$ is the original weight of feature w_{di} (e.g. tf-idf). For example, in a two-view model, each base feature w_{di} would get included twice in the resulting vector representation, once with weight $f(w_{di})p(|z_1|\mathbf{w})$ and once with weight $f(w_{di})p(z_2|\mathbf{w})$. In order to control the effects of noisy features, the resulting vectors are pruned back down to the same number of features as the original, resulting in some loss of base features.

5.2.3.2 Contextualization

There are two main strategies for contextualization in MV-VS:

- **model-based** – As with the basic MV-A vector representation, in MV-VS the non-contextual feature weighting $p(z_1|\mathbf{w})$ can be replaced with a contextualized version $p(z_1|\mathbf{w}, \mathbf{c})$ using either of the contextualization strategies discussed in §5.2.1.1.

- **vector-centroid** – The simplest way to combine MV-VS vectors to form contextual representations is *second-order* vector averaging. Each word w with vector representation $\mathbf{v}^{(w)}$ and context vector set $\{\mathbf{v}^{(c)}|c \in C(w)\}$ can be represented as

$$\mathbf{v}^{(\mathbf{w},\mathbf{c})} = \frac{1}{|C|+1}\left[\mathbf{v}^{(w)} + \sum_{c \in C} \mathbf{v}^{(c)}\right],$$

5.3 Human Evaluation of MV-C Word Representations

5.3.1 Word Representation

The base corpora introduced in §3 are divided into two experimental groups:

1. **Syntax-only** – Words are represented using only the Google n-gram **context features** set. Two versions of this feature set are explored:

   (a) the *common* subset contains all syntactic contexts appearing more than 200 times in the combined corpus, and

   (b) the *rare* subset, containing only contexts that appear 50 times or fewer.

Context Intrusion		
__ is characterized	top of the __	*country to* __
symptoms of __	*of __ understood*	__ or less
cases of __	along the __	__ a year
in cases of __	portion of the __	__ per day
real estate in __	side of the __	__ or more
Word Intrusion		
metal	dues	humor
floral	premiums	ingenuity
nylon	pensions	*advertisers*
what	*did*	delight
ruby	damages	astonishment
Document Intrusion		
Puerto Rican cuisine	Adolf Hitler	History of the Han Dynasty
Greek cuisine	*List of General Hospital characters*	Romance of the Three Kingdoms
ThinkPad	History of France	*List of dog diseases*
Palestinian cuisine	Joachim von Ribbentrop	Conquest of Wu by Jin
Field ration	World War I	Mongolia

Table 5.1: Example questions from the three intrusion tasks, in order of difficulty (left to right, easy to hard; computed from inter-annotator agreement). *Italics* show intruder items.

2. **Syntax+Documents** – Words are represented using a combination of **context features** and Wikipedia **article occurrence** features.

Models trained on the **syntax-only** set are only capable of capturing *syntagmatic* similarity relations, that is, words that tend to appear in similar contexts. In contrast, the **syntax+documents** set broadens the scope of modelable similarity relations, allowing for *paradigmatic* similarity (e.g. words that are topically related, but do not necessarily share common syntactic contexts).

5.3.2 Evaluation Procedure

Our main goal in this work is to find models that capture aspects of the syntactic and semantic organization of word in text that are intuitive to humans. According to the *use theory* of meaning, lexical semantic knowledge is equivalent to knowing the contexts that words appear in, and hence being able to form reasonable hypotheses about the relatedness of syntactic contexts.

Vector space models are commonly evaluated by comparing their similarity predictions to a nominal set of human similarity judgments (Curran, 2004a; Padó and Lapata, 2007; Schütze, 1998a; Turney, 2006). In this work, since we are evaluating models that potentially yield many different similarity scores, we take a different approach, scoring clusters on their semantic and syntactic *coherence* using a *set intrusion* task (Chang et al., 2009a).

In set intrusion, human raters are shown a set of options from a coherent group and asked to identify a single *intruder* drawn from a different group. We ex-

tend intrusion to three different lexical semantic tasks: (1) *context intrusion*, where the top contexts from each cluster are used, (3) *document intrusion*, where the top document contexts from each cluster are used, and (2) *word intrusion*, where the top words from each cluster are used. For each cluster, the top four contexts/words are selected and appended with another context/word from a different cluster.[1] The resulting set is then shuffled, and the human raters are asked to identify the intruder, after being given a short introduction (with common examples) to the task. Table 5.1 shows sample questions of varying degrees of difficulty. As the semantic coherence and distinctness from other clusters increases, this task becomes easier.

Set intrusion is a more robust way to account for human similarity judgments than asking directly for a numeric score (e.g., the Miller and Charles (1991) set) as less calibration is required across raters. Furthermore, the additional cluster context significantly reduces the variability of responses.

Human raters were recruited from *Amazon's Mechanical Turk*. A total of 1256 raters completed 30438 evaluations for 5780 unique intrusion tasks (5 evaluations per task). 2736 potentially fraudulent evaluations from 11 raters were rejected.[2] Table 5.3 summarizes inter-annotator agreement. Overall we found $\kappa \approx 0.4$ for most tasks; a set of comments about the task difficulty is given in Table 5.2, drawn from an anonymous public message board.

[1] Choosing four elements from the cluster uniformly at random instead of the top by probability led to lower performance across all models.

[2] (**Rater Quality**) Fraudulent Turkers were identified using a combination of average answer time, answer entropy, average agreement with other raters, and adjusted answer accuracy.

U1	I just tried 30 of the what doesn't belong ones. They took about 30 seconds each due to thinking time so not worth it for me.
U2	I don't understand the fill in the blank ones to be honest. I just kinda pick one,since I don't know what's expected lol
U3	Your not filling in the blank just ignore the blank and think about how the words they show relate to each other and choose the one that relates least. Some have just words and no blanks.
U4	These seem very subjective to mw. i hope there isn't definite correct answers because some of them make me go [emoticon of head-scratching]
U5	I looked and have no idea. I guess I'm a word idiot because I don't see the relation between the words in the preview HIT - too scared to try any of these.
U6	I didn't dive in but I did more than I should have they were just too easy. Most of them I could tell what did not belong, some were pretty iffy though.

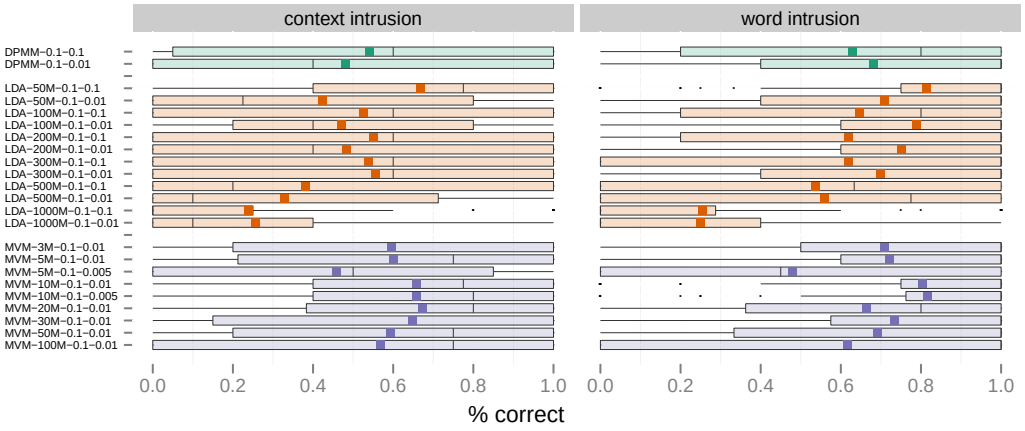
Table 5.2: Sample of comments about the task taken verbatim from a public Mechanical Turk user message board (TurkerNation). Overall the raters report the task to be difficult, but engaging.

5.3.3 Results

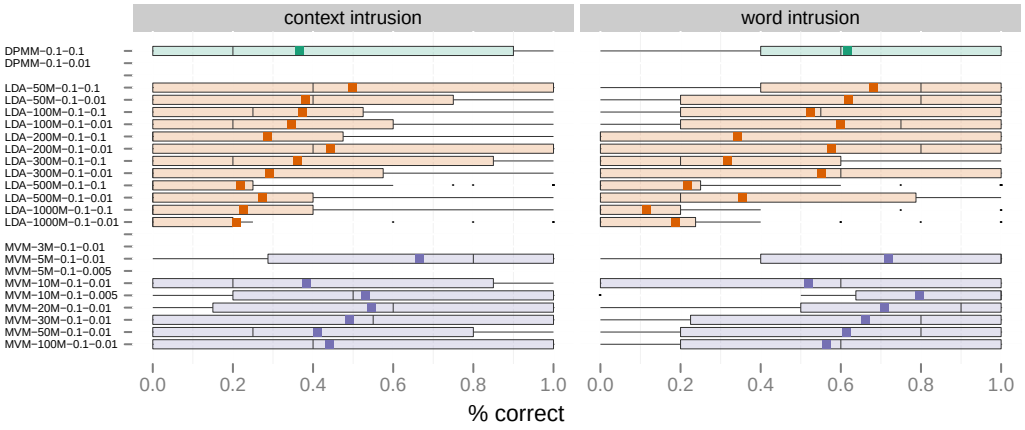
DPMM, MV-A and MV-C models were trained on the **syntax-only** and **syntax+documents** data across a wide range of settings for

$$M \in \{3, 5, 7, 10, 20, 30, 50, 100, 200, 300, 500, 1000\},$$

$\alpha \in \{0.1, 0.01\}$, and $\beta \in \{0.1, 0.05, 0.01\}$ in order to understand how they perform relatively on the intrusion tasks and also how sensitive they are to various parameter

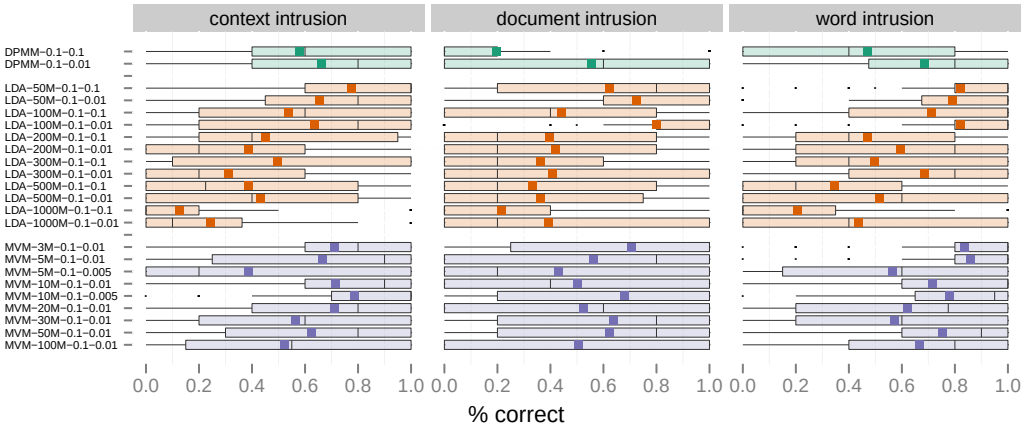


(a) **Syntax-only**, common n-gram contexts.



(b) **Syntax-only**, rare n-gram contexts.

Figure 5.3: (Caption next page)

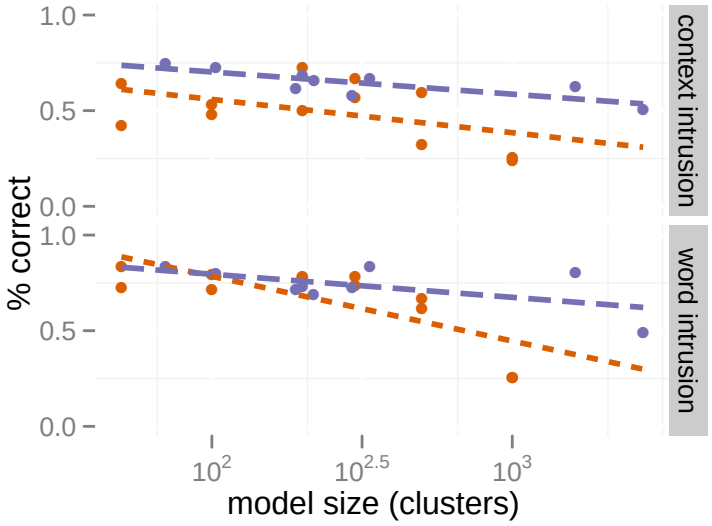


(c) **Syntax+Documents**, common n-gram contexts.

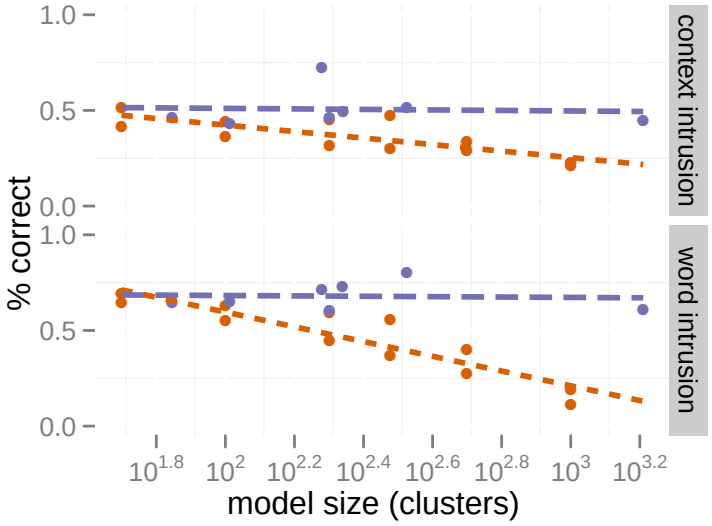
Figure 5.3: Average scores for each model broken down by parameterization and base features. Error bars depict 95% confidence intervals. X-axis labels show **Model-views**-α-β. Dots show average rater scores; bar-charts show standard quantile ranges and median score.

Model	Syntax	Syntax+Documents	**Overall**
DPMM	0.30	0.40	**0.33**
MV-A	0.33	0.39	**0.35**
MV-C	0.44	0.49	**0.46**
Overall	**0.37**	**0.43**	**0.39**

Table 5.3: Fleiss' κ scores for the intrusion detection task across various model and data combinations. Results from MV-C have higher κ scores than MV-A or DPMM; likewise **Syntax+Documents** data yields higher agreement, primarily due to the relative ease of the document intrusion task. Overall refers to the row- or column-marginals (e.g. the *overall* κ for the syntax models, or the overall κ for DPMM across all data types). Fleiss' κ was chosen as it is a standard measure of inter-annotator agreement.



(a) **Syntax-only**, common n-gram contexts.



(b) **Syntax-only**, rare n-gram contexts.

Figure 5.4: Scatterplot of model size vs. avg score for MV-C (dashed, purple) and MV-A (dotted, orange).

settings.[3] MV-A is run on a different range of M settings from MV-C (50-1000 vs 3-100) in order to keep the effective number of clusters (and hence model capacity) roughly comparable.

Models were run until convergence, defined as no increase in log-likelihood on the training set for 100 Gibbs samples. Average runtimes varied from a few hours to a few days, depending on the number of clusters or topics. There is little computational overhead for MV-C compared to MV-A or DPMM with a similar number of clusters.

Overall, MV-C significantly outperforms both MV-A and DPMM (measured as % of intruders correctly identified) as the number of clusters increases. Coarse-grained lexical semantic distinctions are easy for humans to make, and hence models with fewer clusters tend to outperform models with more clusters. Since high granularity predictions are more useful for downstream tasks, we focus on the interplay between model complexity and performance.

5.3.3.1 Syntax-only Model

For common n-gram context features, MV-C performance is significantly less variable than MV-A on both the word intrusion and context intrusion tasks, and furthermore significantly outperforms DPMM (Figure 5.3a). For context intrusion, DPMM, MV-A, and MV-C average 57.4%, 49.5% and 64.5% accuracy respectively; for word intrusion, DPMM, MV-A, and MV-C average 66.7%, 66.1% and 73.6%

[3] We did not compare directly to Cross-cutting categorization, as the Metropolis-Hasting steps required that model were too prohibitively expensive to scale to the Google n-gram data.

accuracy respectively (averaged over all parameter settings). These models vary significantly in the average number of clusters used: 373.5 for DPMM, 358.3 for MV-A and 639.8 for MV-C, i.e. the MV-C model is significantly more granular. Figure 5.4a breaks out model performance by model complexity, demonstrating that MV-C has a significant edge over MV-A as model complexity increases.

For rare n-gram contexts, we obtain similar results, with MV-C scores being less variable across model parameterizations and complexity (Figure 5.3b). In general, MV-A performance degrades faster as model complexity increases for rare contexts, due to the increased data sparsity (Figure 5.4b). For context intrusion, DPMM, MV-A, and MV-C average 45.9%, 36.1% and 50.9% accuracy respectively; for word intrusion, DPMM, MV-A, and MV-C average 67.4%, 45.6% and 67.9% accuracy; MV-C performance does not differ significantly from DPMM, but both outperform MV-A. Average cluster sizes are more uniform across model types for rare contexts: 384.0 for DPMM, 358.3 for MV-A and 391 for MV-C.

Human performance on the context intrusion task is significantly more variable than on the word-intrusion task, reflecting the additional complexity. In all models, there is a high correlation between rater scores and per-cluster likelihood, indicating that model confidence reflects noise in the data.

5.3.3.2 Syntax+Documents Model

With the **syntax+documents** training set, MV-C significantly outperforms MV-A across a wide range of model settings. MV-C also outperforms DPMM for word and document intrusion. For context intrusion, DPMM, MV-A, and MV-C av-

erage 68.0%, 51.3% and 66.9% respectively;[4] for word intrusion, DPMM, MV-A, and MV-C average 56.3%, 64.0% and 74.9% respectively; for document intrusion, DPMM, MV-A, and MV-C average 41.5%, 49.7% and 60.6% respectively. Qualitatively, models trained on **syntax+document** yield a higher degree of paradigmatic clusters which have intuitive thematic structure. Performance on document intrusion is significantly lower and more variable, reflecting the higher degree of world knowledge required. As with the previous data set, performance of MV-C models trained on **syntax+documents** data degrades less slowly as the cluster granularity increases (Figure 5.5).

One interesting question is to what degree MV-C *views* partition syntax and document features versus MV-A topics. That is, to what degree do the MV-C views capture purely syntagmatic or purely paradigmatic variation? We measured *view entropy* for all three models, treating syntactic features and document features as different class labels. MV-C with $M = 50$ views obtained an entropy score of 0.045, while MV-A with $M = 50$ obtained 0.073, and the best DPMM model 0.082.[5] Thus MV-C views may indeed capture pure syntactic or thematic clusterings.

5.3.4 Discussion

As cluster granularity increases, we find that MV-C accounts for feature noise better than either MV-A or DPMM, yielding more coherent clusters. (Chang

[4]High DPMM accuracy is driven by the low number of clusters: 46.5 for DPMM vs. 358.3 for MV-A and 725.6 for MV-C.

[5]The low entropy scores reflect the higher percentage of syntactic contexts overall.

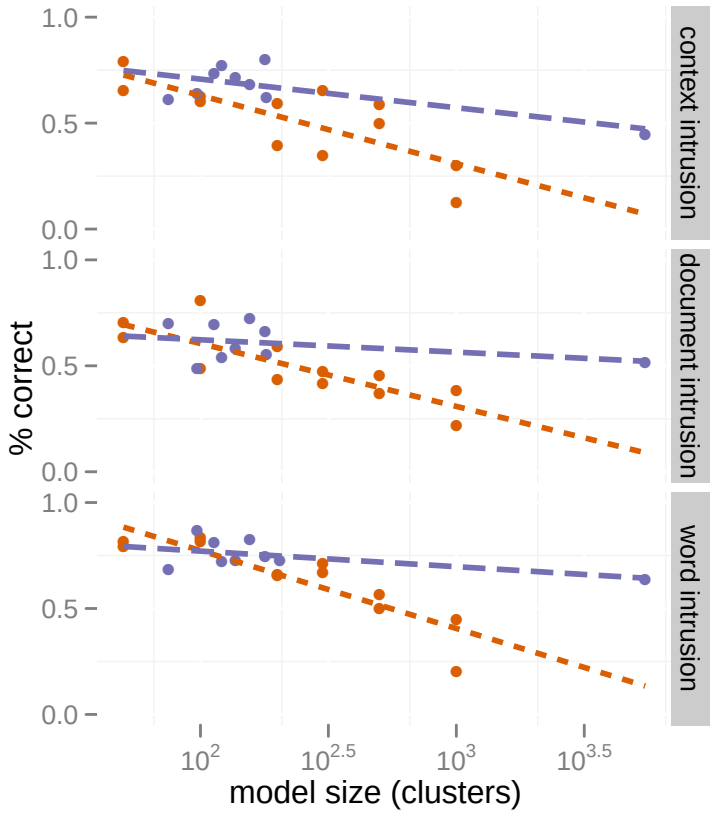


Figure 5.5: Scatterplot of model size vs. average score for MV-C (dashed, purple) and MV-A (dotted, orange); **Syntax+Documents** data.

et al., 2009a) note that MV-A performance degrades significantly on a related task as the number of topics increases, reflecting the increasing difficulty for humans in grasping the connection between terms in the same topic. This suggests that as topics become more fine-grained in models with larger number of topics, they are less useful for humans. In this work, we find that although MV-C and MV-A perform similarity on average, MV-C clusters are significantly more interpretable than MV-A clusters as the granularity increases (Figures **??** and 5.5). We argue that models capable of making such fine-grained semantic distinctions are more desirable.

The results presented in the previous two sections hold both for unbiased cluster selection (e.g. where clusters are drawn uniformly at random from the model) *and* when cluster selection is biased based on model probability (results shown). Biased selection potentially gives an advantage to MV-C, which generates many more small clusters than either MV-A or DPMM, helping it account for noise.

5.4 Lexical Semantic Evaluation

This section compares the performance of MV-A and MV-VS to the baseline single-prototype model (VS) on various lexical semantic tasks. In order to simplify presentation of the results, MV-C was excluded, as it does not significantly outperform MV-A on these tasks.

5.4.1 McRae

On the McRae categorization norms, for both the category naming and exemplar generation subtasks, the category-constituent representation significantly outperforms the category-label representation across all models and settings. Furthermore, this effect is stronger for category naming than for exemplar generation.

In general, MV-VS outperforms MV-A, with performance similar to, and in some cases better than, VS. Furthermore, MV-VS-V and MV-VS-M are nearly identical in terms of performance.

5.4.1.1 Category Naming

On the category naming subtask, the best performing results are found using the category-constituent representation, **contexts** training data, and either VS or MV-VS with low number of views (fewer than 100; MRR ≈ 0.498; Figure 5.6). This result may potentially indicate that local syntactic structure is more predictive of category labels. Using **unigrams** features, MV-VS with a high number of views (100-1000) significantly outperforms VS (MRR ≈ 0.426 vs. MRR ≈ 0.361; although the results are worse than when using **contexts** features).

For MV-A, across all datasets, MRR increases as the number of views is increased. However for MV-VS, MRR increases only for **all** and **unigrams** features, but decreases for **contexts** and **wikipedia** features.

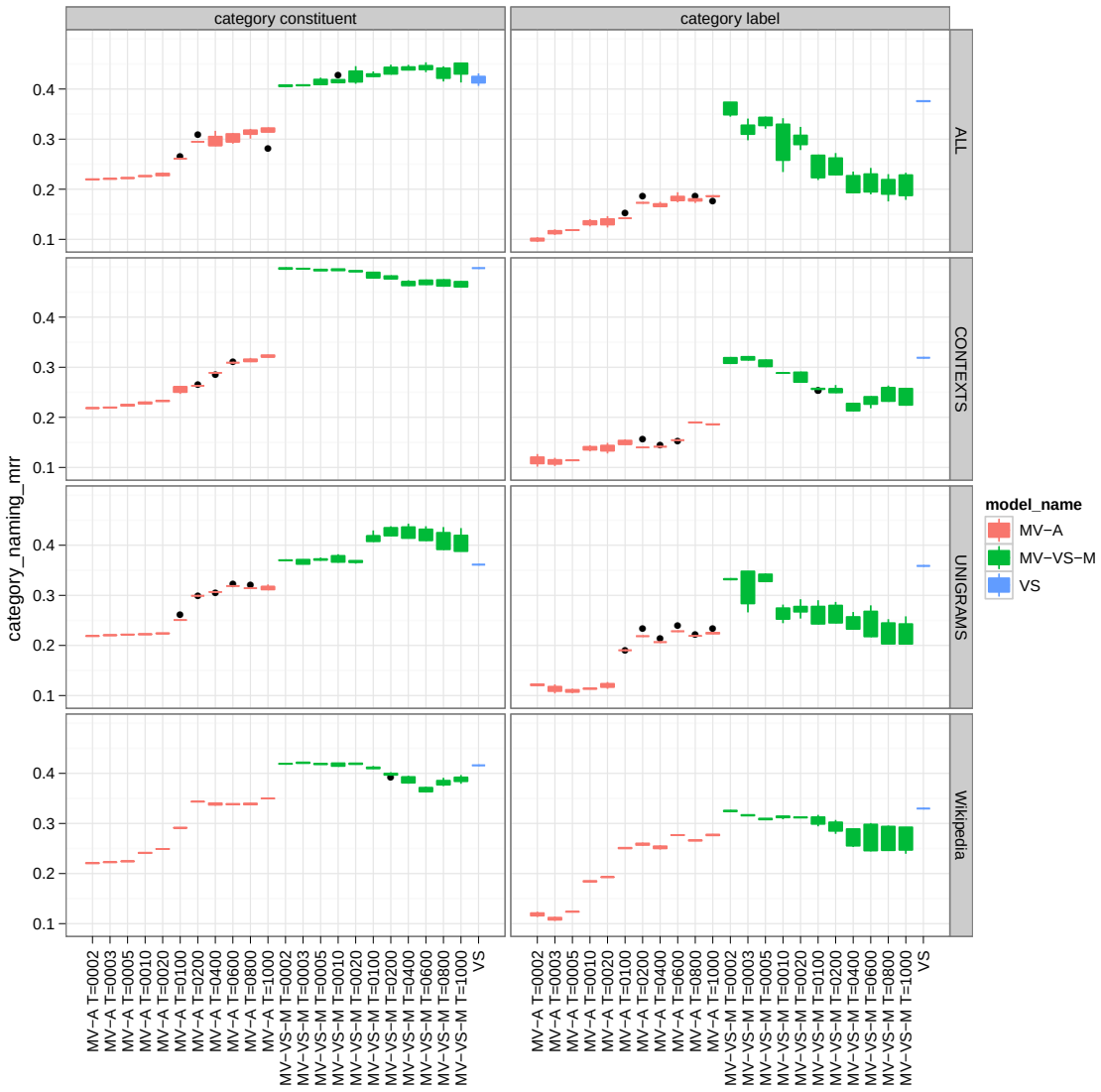


Figure 5.6: **McRae** mean-reciprocal rank (MRR) scores for the category naming task (§3.2.3). Columns break down scores by the category-label or category-constituent representation and rows break down scores by source data set.

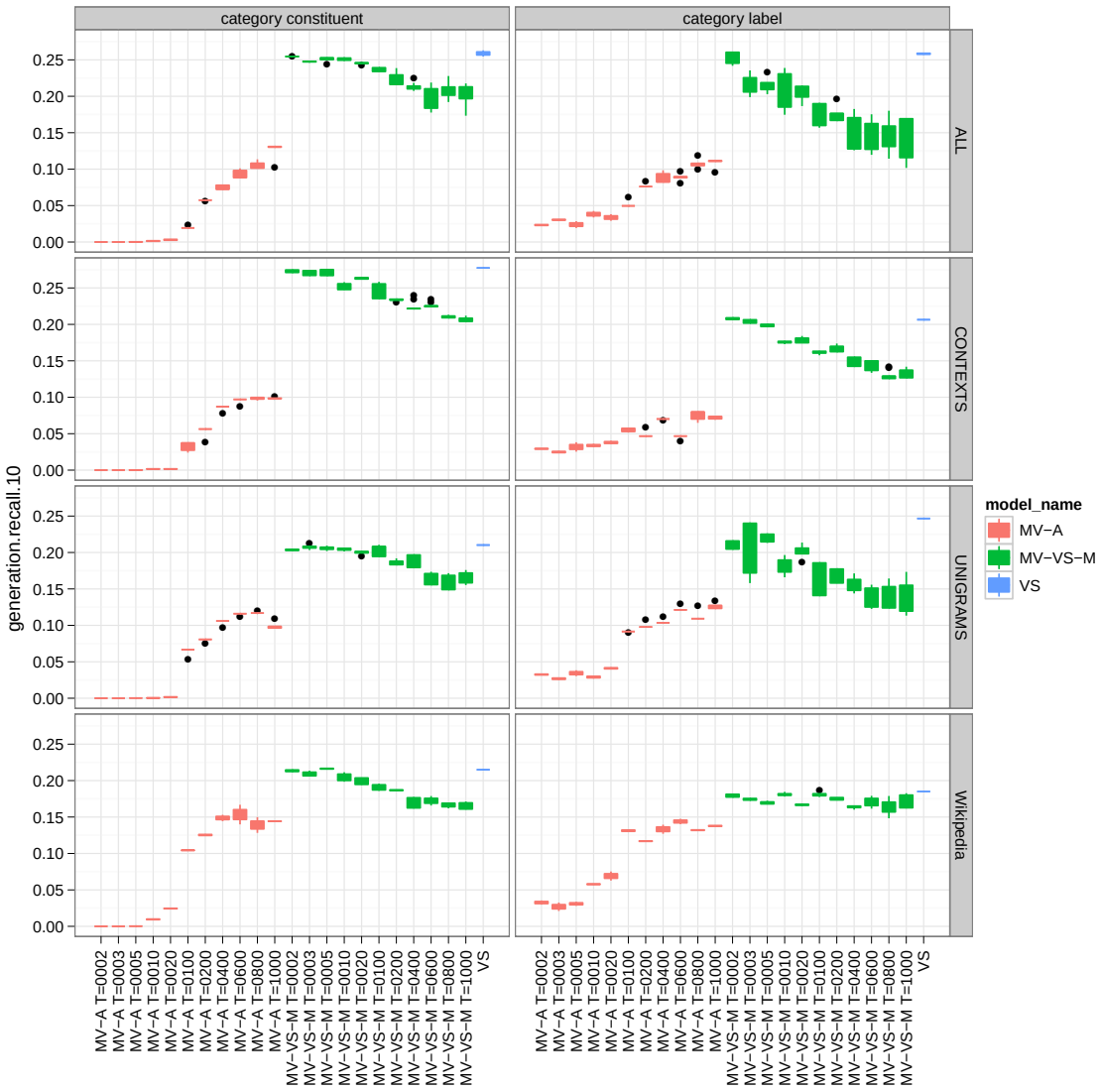


Figure 5.7: **McRae** recall10 scores for the exemplar generation (§3.2.3). Columns break down scores by category-label or category-constituent representation and rows break down scores by source data set.

5.4.1.2 Exemplar Generation

For the exemplar generation task, again the category-constituent representation generally outperforms the category-label representation across all models, except in the case of VS with **all** features, where recall is the same (Figure 5.7). The best MV-VS recall does not differ significantly from the VS baseline (Recall ≈ 0.273 vs. Recall ≈ 0.277 using **contexts** features). As the number of views increases, however, MV-VS recall falls significantly across all cases. In terms of features, **contexts** again yield the best recall (0.277), followed closely by **all** (0.252).

5.4.1.3 Discussion

Unlike in the case of the word-occurrence models (§4.5.4), the category-constituent representation in the word-type models is found to significantly outperform the category-label representation for category naming. It is interesting that performance is sensitive to category representation for category naming, but is significantly less so for exemplar generation (MV-VS is the most sensitive to the underlying category representation). Given the fact that the category-constituent representation makes use of the remaining category exemplars, a less surprising result would be for the category-constituent representation to perform better.

5.4.2 BLESS

5.4.2.1 Overall GAP

Averaged over all BLESS relation types (**attribute**, **coordinate**, **event**, **hypernym**, and **meronym**), the best performance is achieved using the baseline VS

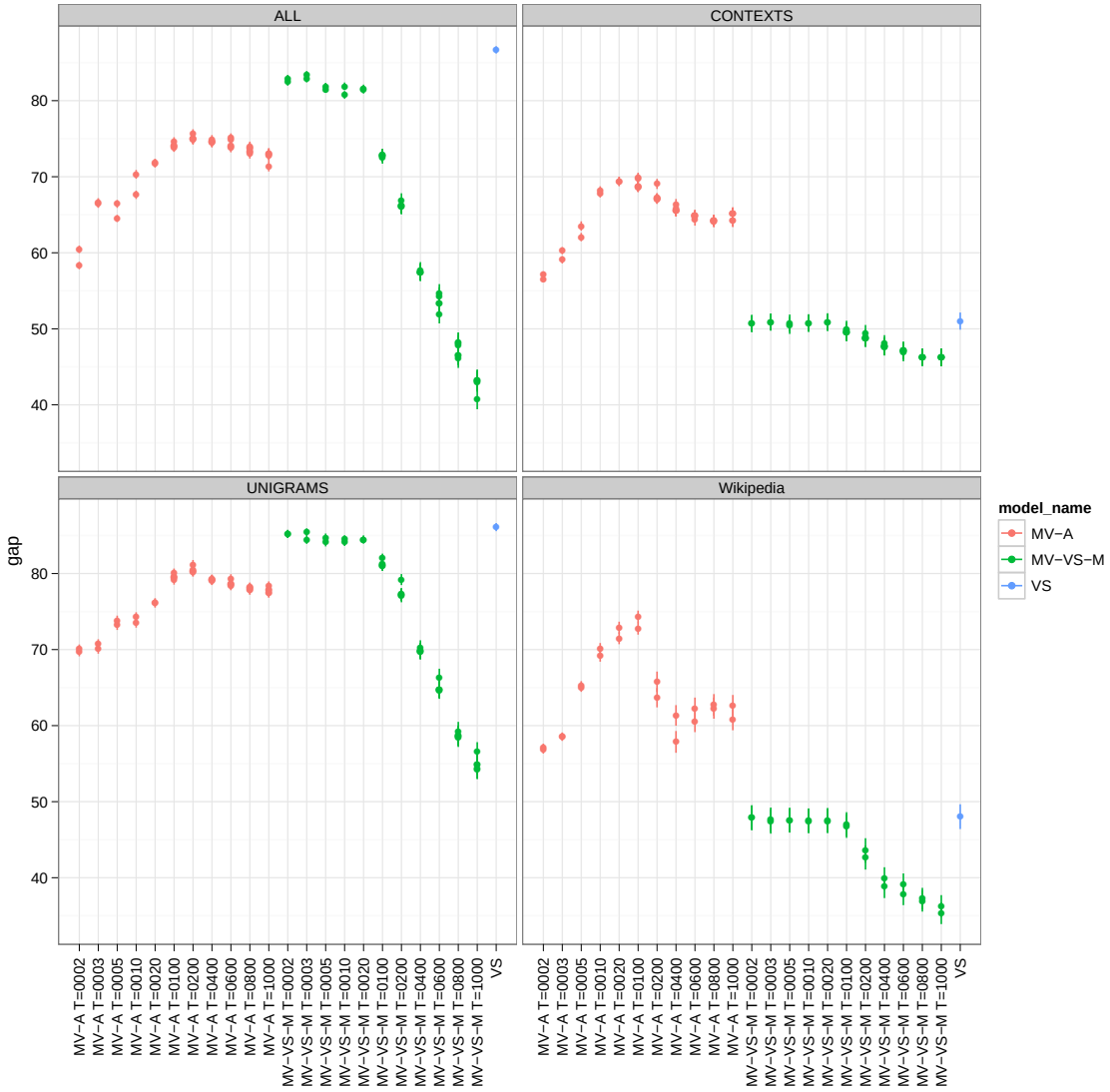


Figure 5.8: **BLESS** GAP broken down by base features. In general, models trained using **unigrams** features perform the best.

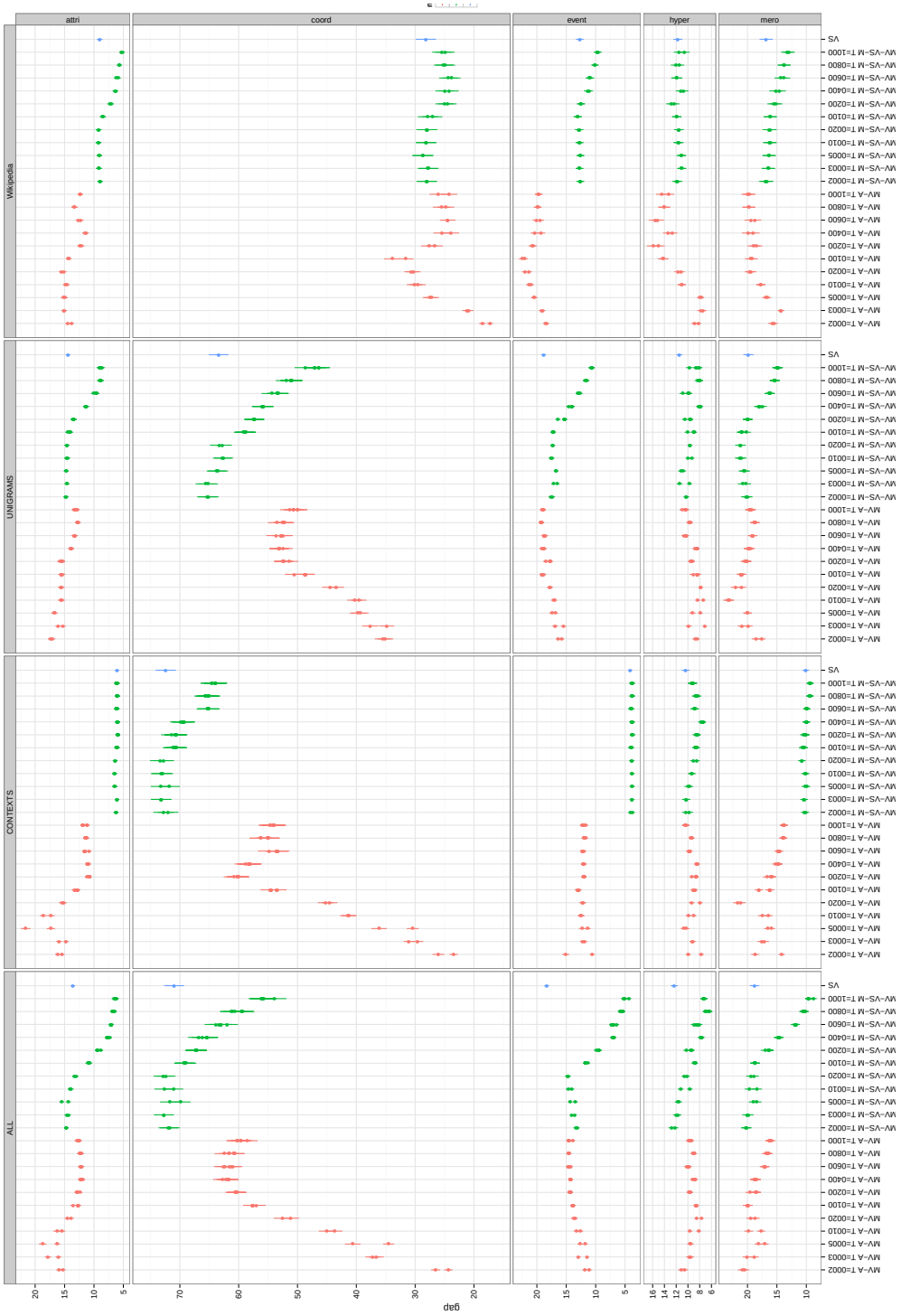


Figure 5.9: **BLESS** GAP broken down by base features and relation.

model trained on either **unigrams** or **all** features (GAP $\approx$ 86.68; Figure 5.8). In contrast, the best performing MV-VS models achieve GAP $\approx$ 83.16, and the best performing MV-A models achieve GAP $\approx$ 80.48.

Moving to **contexts** or **wikipedia** features, however, the best performing MV-A beats both MV-VS and VS (**contexts**: GAP $\approx$ 69.38 for MV-A $T = 20$ vs. GAP $\approx$ 50.93 for MV-VS $T = 3$ and GAP $\approx$ 51.0 for VS; **wikipedia**: GAP $\approx$ 73.53 for MV-A $T = 100$ vs. GAP $\approx$ 47.87 for MV-VS $T = 2$ and GAP $\approx$ 48.03 for VS). MV-A performance peaks around 200-400 views, while MV-VS performance drops off significantly after 100 views.

Finally, as with the McRae categorization task, there is no significant difference between MV-VS-V and MV-VS-M.

5.4.2.2 Per-Relation GAP

GAP scores broken down by feature type and relation are summarized in Figure 5.9.

- **attribute** relations: MV-A with fewer than 200 views performs the best across all data sets (peaking around 20 views), beating MV-VS and VS (GAP $\approx$ 17.21 vs. GAP $\approx$ 14.89 and GAP $\approx$ 14.4). Using **all** or **contexts** features slightly outperforms using **unigrams** or **wikipedia** features.

- **coordinate** relations: MV-VS with a low number of views (< 10) outperforms the baseline VS on **all**, **contexts** and **unigrams** features, however these results are not statistically significant (GAP $\approx$ 73.17 vs. GAP $\approx$ 72.42). Us-

ing **all** or **contexts** features yields the highest coordinate term GAP across all models. Features from **wikipedia** yield the lowest coordinate term GAP (improving GAP on other relations). Across all relations coordinate term GAP is the highest.

- **event** relations: The highest GAP scores on event relations are obtained by MV-A trained on **wikipedia** features (GAP ≈ 22.28 vs. GAP ≈ 13.75 for MV-VS and GAP ≈ 12.71 for baseline VS). Indeed, MV-A outperforms both VS and MV-VS on all feature types except **all**, where VS outperforms all other models (GAP ≈ 18.33).

- **hypernym** relations: Similar to event relations, the best GAP scores for hypernym relations are achieved by MV-A on the **wikipedia** feature set (GAP ≈ 15.44 vs. GAP ≈ 12.45 for MV-VS and GAP ≈ 12.32 for baseline VS).

- **meronym** relations: The best GAP performance for meronym relations is achieved by MV-A on the **unigrams** feature set (GAP ≈ 21.17 vs. GAP ≈ 17.47 for MV-VS and GAP ≈ 18.84 for baseline VS). In general for meronyms, the model-based approaches outperform the baseline VS model, except in the case of **unigrams** features.

Overall, taking these results together, both the feature space and model type contribute significantly to the types of relations recalled. Compared to other feature types, **wikipedia** features are significantly worse at yielding coordinate terms, but yield correspondingly more lexical items following the other relation types.

When using **contexts** data as a vector space (e.g. VS and MV-VS), GAP scores stay near zero for **attri**, **event** and **mero** relations. In other words, almost all recalled lexical items follow a **coordinate** relation. This makes sense, given that vectors which share many local syntactic features most likely have *syntagmatic* relations. Hence, in order to generalize to other relation types using purely **contexts** data, higher order model structure (e.g. that found in MV-A) is necessary.

5.4.3 Lexical Substitution

For the lexical substitution tasks, we follow the evaluation procedure of Dinu and Lapata (2010), where only the target word is represented in context. Leaving the representations for the paraphrase candidates uncontextualized resulted in significantly higher performance across all task settings.

5.4.3.1 LexSub07

On the LexSub07 evaluation set, the best performing model is the baseline VS using 1 word context windows (vector centroid) trained using **all** features (GAP $\approx$ 48.27) followed by **unigrams** features (GAP $\approx$ 47.32; Figure 5.10). The corresponding MV-VS models achieve GAP $\approx$ 46.83 for **all** features and GAP $\approx$ 46.77 for **unigrams** features. In general across all models evaluated, using **all** features yields the highest GAP scores, indicating potential for further improving performance by including more features.

In the case of **wikipedia** features, the best performing MV-VS models achieve higher GAP scores than the baseline VS (GAP $\approx$ 44.23 with $T = 2$ vs. GAP $\approx$

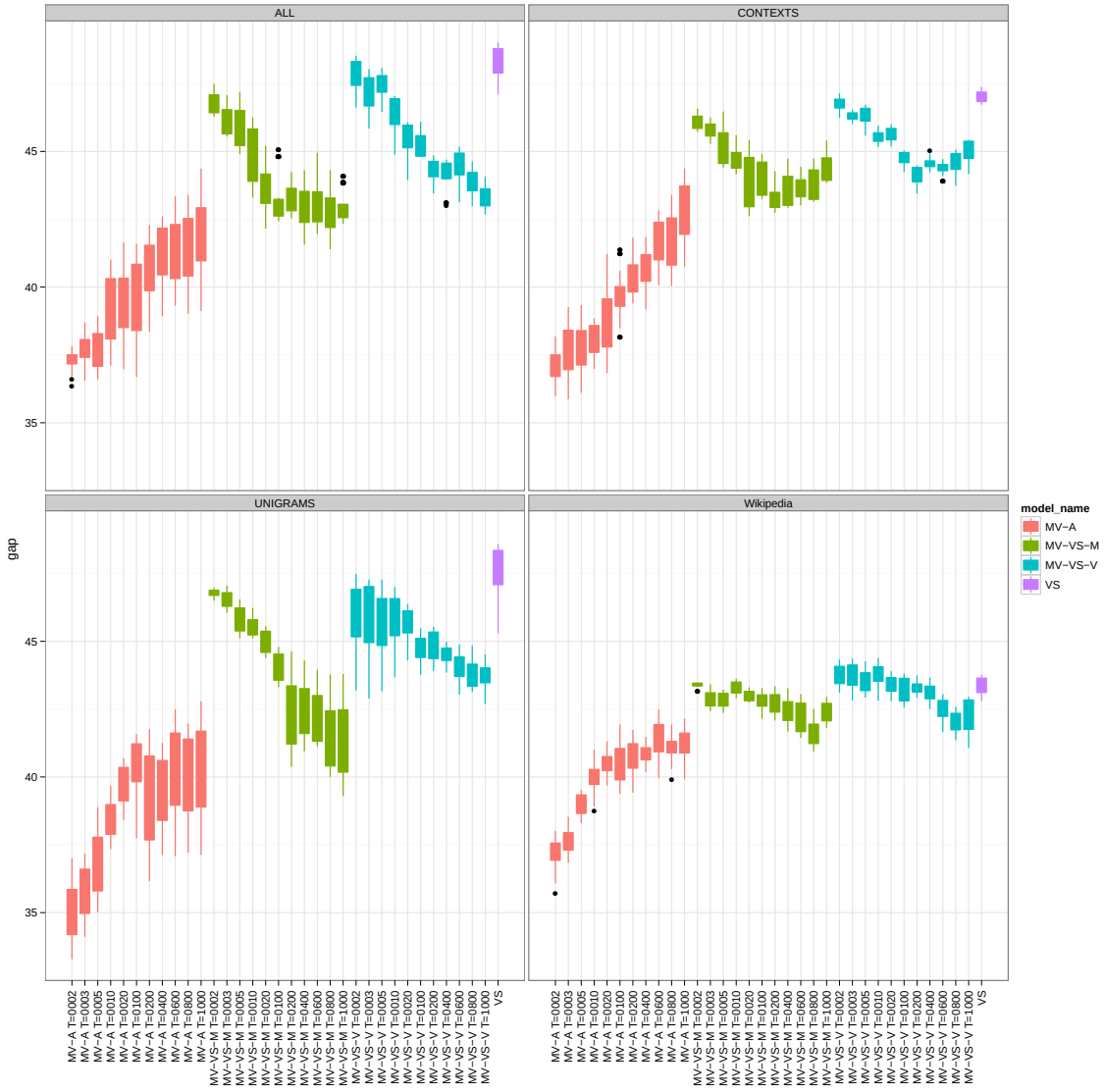


Figure 5.10: **LexSub07** GAP scores broken down by base feature set.

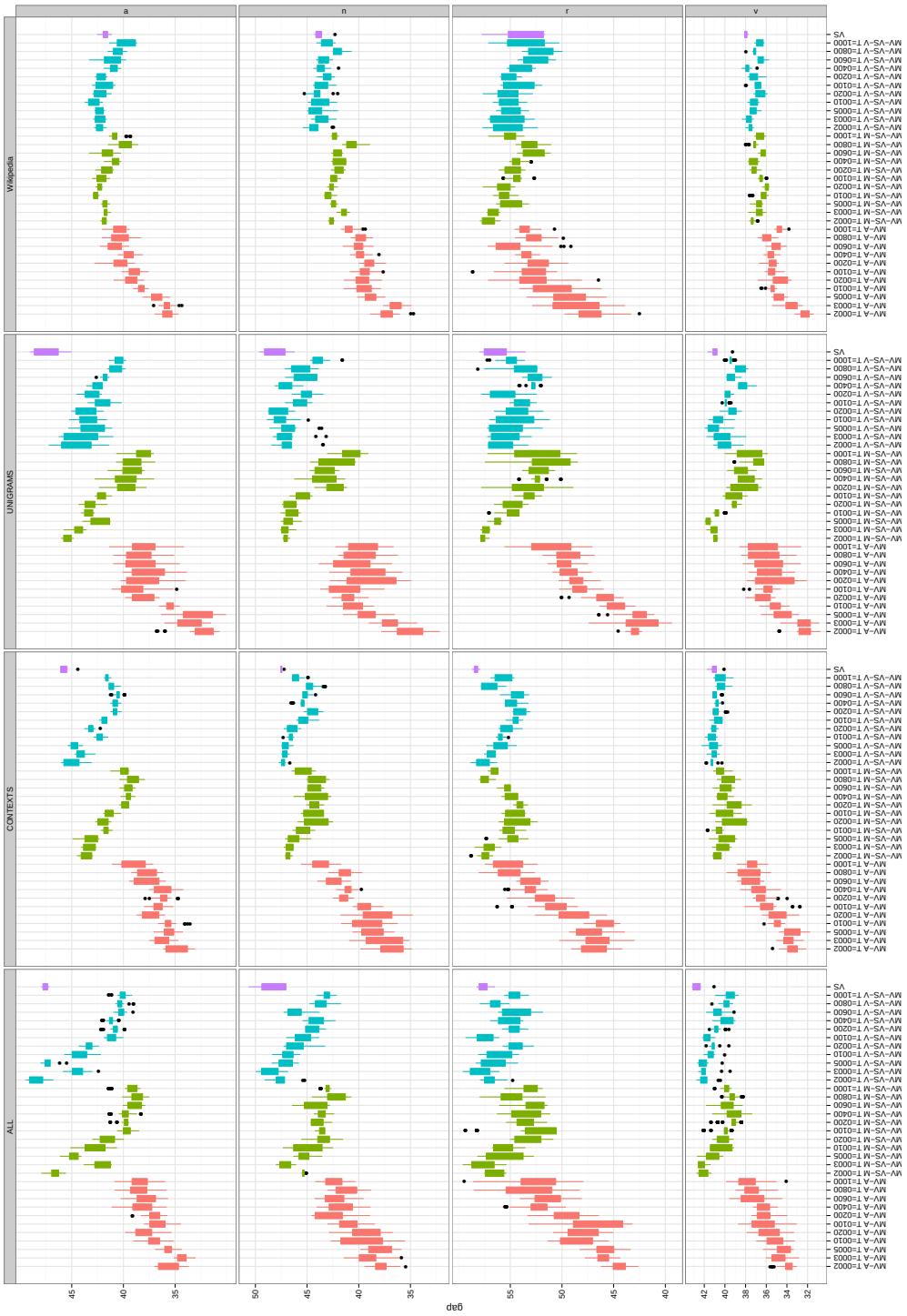


Figure 5.11: **LexSub07** GAP results broken down by source feature set (columns) and part of speech (rows).

43.57). However, using **wikipedia** features alone results in the worst performance overall.

MV-VS performance decreases as the number of views is increased, whereas MV-A performance increases. Unlike in Ó Séaghdha and Korhonen (2011), using the experimental setup here, there was no significant difference between the **DL10** and **OK11** contextualization strategies. Unlike on the McRae or BLESS test sets, in LexSub07, there is some substantial difference in the performance of MV-VS-M and MV-VS-V. MV-VS-V GAP scores are significantly less variable, and performance does not degrade as much as the number of views is increased (this effect is most pronounced for the **unigrams** feature set).

Finally, the GAP scores achieved by the standard vector space model using first-order centroid contextualization outperform many more complex approaches presented in the literature for LexSub07, including both models presented by Thater et al. (2010) as well as the W5 model from Ó Séaghdha and Korhonen (2011) (which in turn was shown to outperform Dinu and Lapata (2010)). Given the inherent noise in the human evaluation data (Table 3.6), it is most likely that any additional improvement above GAP ≈ 50.0 is due to random chance.

Figure 5.11 shows GAP scores broken down by the part-of-speech of the target word. For adjectives (a) and adverbs (r), MV-VS with a low number of views $T = 2$ significantly outperforms the baseline VS model using **all** features (GAP $\approx$ 49.33 vs. GAP $\approx$ 47.59 for adjectives and GAP $\approx$ 59.12 vs. GAP $\approx$ 57.48 for adverbs). Indeed, nouns are the only part of speech type for which VS performance is similar to MV-VS (GAP ≈ 48.61 for VS vs. GAP ≈ 48.55 for MV-VS). In terms

of absolute performance, adverb GAP is the highest across all models, while verb GAP is the lowest.

5.4.3.2 TWSI1

In general, the results on the *TWSI1* lexical substitution task are similar to those obtained for *LexSub07*. However, the best performing model on TWSI1 is MV-VS with $T = 2$ using **all** features (GAP ≈ 61.29 vs. GAP ≈ 60.58 for the VS baseline). Unlike in LexSub07, GAP performance on the **wikipedia** feature set is not significantly lower than the other feature sets.

5.5 Discussion

This chapter introduced three *multi-view* models of lexical semantics: (1) MV-A where word features are assigned across multiple views using LDA, (2) MV-C, an extension to MV-A where word features are further clustered within each view, capturing multiple lexical similarity relations jointly in the same model, and (3) MV-VS, a vector-space model that accounts for the "senses" (view-assignments) of individual word features. As demonstrated in the empirical results, MVM naturally captures both *syntagmatic* and *paradigmatic* notions of word similarity. Furthermore MVM-based models perform favorably compared to other generative lexical semantic models on a set of human evaluations: concept categorization, multi-relational word-similarity and lexical substitution.

.

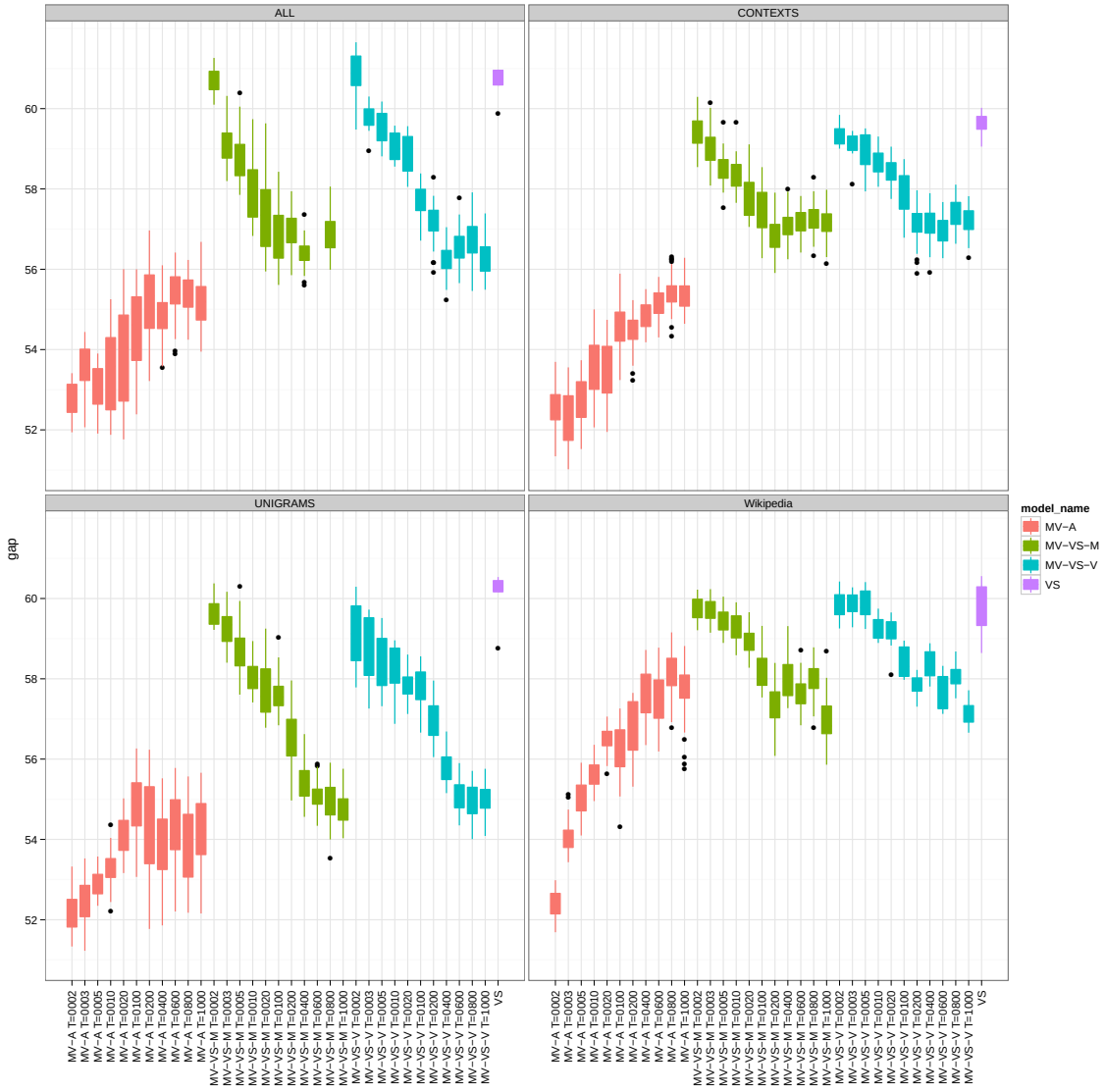


Figure 5.12: **TWSI1** GAP scores broken down by base feature set.

Chapter 6

Future Work

6.1 Multi-Prototype and Tiered Clustering

6.1.1 Scaling Representational Capacity

The success of the *combined* multi-prototype approach (§4.5.2; combining prototypes across multiple clustering scales) indicates that the optimal number of clusters may vary per word. The DPMM-based model addresses this directly, however there are several other principled approaches to automatically assessing clustering capacity:

1. Kilgarriff (2004) demonstrate that word sense frequency distributions are Zipfian word-frequency (Zipf, 1935). Hence, simply allocating representational capacity in the form of additional prototypes proportional to the total number of occurrences may yield optimal meaning representations, trading off expressivity and robustness.

2. The two-parameter Pitman-Yor generalization of the Dirichlet Process (Pitman and Yor, 1997) yields power-law distributed cluster sizes,

$$P(z_i = k|z_{-i}) = \begin{cases} \frac{n_k^{-i} - d}{\sum_j n_j - 1 + \alpha} & \text{if } n_k > 0 \\ \frac{\alpha + dK}{\sum_j n_j - 1 + \alpha} & k \text{ is a new class.} \end{cases} \tag{6.1}$$

with rate proportional to the free parameter d. Since naturally occurring sense *frequencies* are also roughly power-law distributed (Kilgarriff, 2004), such a model may prove to be a better fit representationally.

6.1.2 Deeper Tiered Structure

The basic tiered clustering model (§4.3.1) can be extended with additional background tiers, allocating more expressivity to model background feature variation. This class of models covers the spectrum between a pure topic model (all background tiers) and a pure clustering model and may be reasonable when there is believed to be more background structure (e.g. when jointly modeling all verb arguments). Furthermore, it is straightforward to extend the model to a two-tier, two-clustering structure capable of additionally accounting for commonalities *between* arguments.

6.1.3 Dense Feature Selection via Bayesian Co-clustering

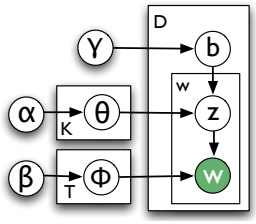
The textual features employed when clustering word occurrences are high-dimensional and sparse and hence noisy. Feature selection and weighting methods like those proposed in the previous sections address the issue of noise, but do not help combat sparsity, and hence many occurrences can end up with few activated features when using feature selection. However, by performing simultaneous dimensionality reduction and feature selection, both issues can be addressed in a coherent framework. This section outlines a simple Bayesian co-clustering approach for simultaneously reducing feature dimensionality and clustering data and shows

how it can be combined with the tiered clustering model.

Co-clustering procedures simultaneously find clusterings of both the rows and the columns of the data matrix, reducing feature dimensionality while grouping data points. Shan and Banerjee (2010) introduce a Bayesian co-clustering approach based on LDA that allows mixed-membership in both the row and column clustering.

One potential simplification of Shan and Banerjee (2010)'s model is to only perform overlap clustering on the features simultaneously with partitioned clustering on the data. The following Bayesian *dense clustering* model clusters documents based on their topic membership proportions:

$$
\begin{array}{rcll}
\boldsymbol{\gamma}|\boldsymbol{\gamma}_0 & \sim & \text{Dirichlet}(\boldsymbol{\gamma}_0), & \text{(cluster proportions} \\
\boldsymbol{\theta}_k|\boldsymbol{\alpha} & \sim & \text{Dirichlet}(\boldsymbol{\alpha}), \quad k \in K, & \text{(topic proportions)} \\
\boldsymbol{\phi}_t|\boldsymbol{\beta} & \sim & \text{Dirichlet}(\boldsymbol{\beta}), \quad t \in T, & \text{(topics)} \\
b_d|\boldsymbol{\gamma} & \sim & \text{Mult}(\boldsymbol{\gamma}), \quad d \in D, & \text{(cluster indicator)} \\
z_{i,d}|\boldsymbol{\theta}_d, b_d & \sim & \text{Mult}(\boldsymbol{\theta}_{b_d}), \quad i \in |\mathbf{w}_d|, & \text{(topic indicator)} \\
w_{i,d}|\boldsymbol{\phi}_{z_{i,d}} & \sim & \text{Mult}(\boldsymbol{\phi}_{z_{i,d}}), \quad i \in |\mathbf{w}_d|, & \text{(words)}
\end{array}
$$



In this model K groups of documents share the same topic proportions $\boldsymbol{\phi}_k$ (i.e. cluster centroids), corresponding to hard-clustering. This model reduces to LDA when $K \rightarrow D$, i.e. each document is assigned to its own cluster, and hence is more computationally efficient than LDA, despite performing clustering and topic-modeling jointly.

Combining the dense clustering model with the tiered clustering model would yield a coherent framework for joint dimensionality reduction, feature selection and clustering, i.e. *dense* feature-selective clustering.

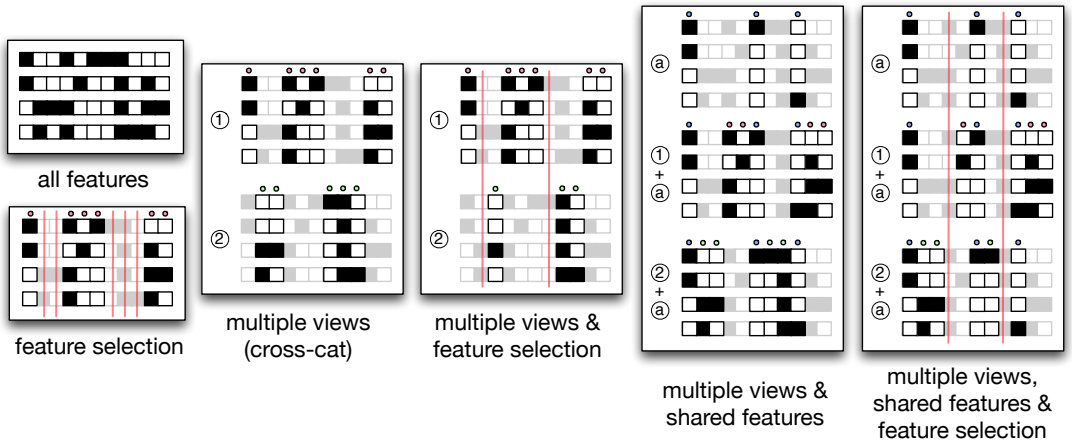


Figure 6.1: Progression of proposed feature selection and multi-view models. Horizontal vectors indicate data; circled numbers and letters represent disparate views; grayed boxes indicate features not present in that particular view; and vertical lines represent features removed from all views. Clustering occurs separately within each view. In the case of shared feature views, features assigned to view (a) are present in all views.

6.2 Multi-View Models

MV-C uses LDA to discover overlapping subsets of features with coherent clusterings, allowing features to be naturally shared between subsets. Cross-cat performs a similar multiple clustering procedure, but features can only be assigned to a single view, limiting its ability to model real-world data. This section derives a set of extensions to Cross-cat building on the *Indian Buffet Process* primitive (Griffiths and Ghahramani, 2006) for multiple assignment, overcoming this limitation.

These models are aimed at overcoming the main limitation of cross-cat, allowing informative features to be shared by several views. The first extension, multiple-views with *shared features*, allows each view to inherit a set of shared features in addition to its view-specific features. The second extension, *factorial*

feature allocation (FFA), puts the entire binary feature assignment matrix $\mathbf{Z}$ under the control of the model, treating it as a random variable. Figures 6.1, 6.2, and 6.3 summarize the various model combinations considered.

6.2.1 Shared Feature Partitions

The first novel extension of cross-cat adds an additional *shared* view that specifies features conserved across all views (Figure 6.1). This puts pressure on the model to identify the most information / most generic features to conserve across clusterings. The shared features themselves do not constitute a separate clustering, and hence do not necessarily need to yield good views on their own. The remaining view-specific features capture the individual idiosyncrasies of each clustering.

The shared feature model is capable of identifying features that contribute to multiple clusterings of the data and hence may find exactly the features that characterize the strongest sense distinctions. For example, features that contribute both to syntactic sense clustering and topical sense clustering. Thus the shared feature model can be viewed a form of *robust clustering*, finding the commonalities between an ensemble of orthogonal clusterings.

Recalling the notation from §2.3.1 The shared view is encoded using an additional random binary vector $\mathbf{u}$ with one entry per feature, indicating whether that feature should be included in all views or not. The resulting construction for $\mathbf{Z}$ is then

$$[\mathbf{Z}]_{f,m} = \begin{cases} 1 & \tilde{\mathbf{z}}_f = m \\ u_f & \text{otherwise.} \end{cases} \tag{6.2}$$

where, again $\tilde{\mathbf{z}} \sim \text{CRP}(\alpha)$, and e.g.,

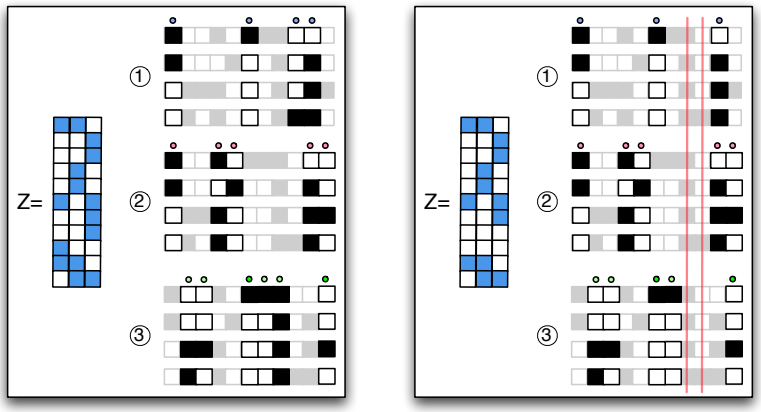
$$u_f|\mu_f \sim \text{Bernoulli}(\mu_f) \tag{6.3}$$

$$\mu_f|\xi \sim \text{Beta}(\xi). \tag{6.4}$$

A similar result could be realized by reserving one cluster in $\tilde{\mathbf{z}}$ to indicate whether the feature is shared or not, however the likelihood structure of this model may cause the sampler not to mix well. However, it may be possible to implement this model using the *colored stick-breaking process* which allows for both exchangeable and non-exchangeable partitions, improving efficiency (Green, 2010).

From a data-analytic perspective this model is interesting because the shared features may capture some intuitive basic structure specific to the particular word, e.g., some notion of the underlying metaphor structure of *line* independent of topical variation.

Klein and Murphy (2001) find no psychological evidence for shared structure linking different senses of polysemous words, indicating that the shared structure model may not perform well relative to the other models proposed here. However, their experiments were not focused on fine-grained sense distinctions such as those present in WordNet, and furthermore this does not necessarily indicate that such models are not applicable to lexical semantics: when deriving occurrence features from raw text, it is expected that there is some feature overlap attributable to the "background" meaning of the word.



factorial feature allocation

factorial feature allocation
+ feature selection

Figure 6.2: Factorial feature allocation model. Horizontal vectors indicate data; circled numbers represent disparate views; grayed boxes indicate features not present in that particular view; and vertical lines represent features removed from all views. The $F \times K$ dimensional matrix $\mathbf{Z}$ specifies what features are present in what view.

6.2.2 Factorial Feature Allocation

Factorial feature allocation puts the full feature-to-view map $\mathbf{Z}$ under the control of the model (Figure 6.2). With FFA each feature is assigned to some subset of the available views, with some probability. The *Indian Buffet Process* provides a suitable nonparametric prior for FFA, where draws are random binary matrices with a fixed number of rows (features) and possibly an infinite number of columns (Griffiths and Ghahramani, 2006). Note that in our case the "latent" feature dimensions inferred by the IBP correspond to feature views in the original clustering problem.

$$\mathbf{Z}|\theta_{\mathbf{Z}} \sim \text{IBP}(\theta_{\mathbf{Z}}) \tag{6.5}$$

which yields feature-to-view assignments where each feature occurs in $A_f \sim \text{Poisson}(\theta)$ views ($\mathbb{E}[A_f] = \theta$), and the total number of views $M \sim \text{Poisson}(\theta H_{|\mathbf{w}|})$[1]

The main benefit of factorial feature allocation over the simpler models is that features can be shared arbitrarily between views, with the IBP specifying only a prior on the *number* of features active in any one view. Concretely, this allows the model to simultaneously represent the most probable clusterings using $\mathbb{E}[\theta F]$ features. Since M is low for most applications considered, factorial feature allocation is not significantly more complex computationally than the shared feature model.

Finally, I propose to explore to what extent to which topic models are similar to factorial feature allocation; at a high level, factorial feature allocation can be viewed as a type of topic model where each topic has only a single word/feature, and may be related to the class of *Focused Topic Models* (Williamson et al., 2010). Exploring this duality should lead to more efficient sampling methods for FFA, as well as topic models better able to capture latent feature structures. Also, it would allow the development of FFA models with latent hierarchical structure, based on e.g. the nested Chinese Restaurant Process (Blei et al., 2003a), labeled LDA (Ramage et al., 2009a) or the Kingman's coalescent (Teh et al., 2007).

6.2.3 Joint Factorial Feature and Data Allocation

The dual problem to feature selection is determining data relevance, i.e. removing outliers or irrelevant data points. Previous lexical semantic models such as

[1] $H_n \stackrel{\text{def}}{=} \sum_{i=1}^{n} \frac{1}{i}$ is the nth harmonic number.

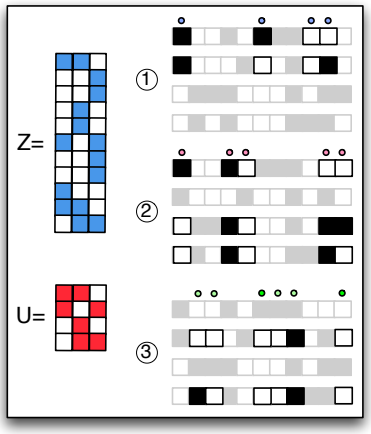


Figure 6.3: Factorial feature and data allocation model. The $F \times K$ dimensional matrix $\mathbf{Z}$ specifies what features are present in what view; the $D \times K$ dimensional matrix $\mathbf{U}$ specifies what data points are allocated to each view.

Clustering by Committee use ad-hoc criteria for improving robustness to outliers (Pantel, 2003); in Statistics, outliers are treated in density estimation using robust distributions, e.g. Laplace (Cord et al., 2006). In addition to studying the applicability of *background cluster* models such as the colored stick-breaking process to clustering word occurrences (Green, 2010), I propose extending the FFA model described in §6.2 to jointly model feature and data allocation (FFDA; Figure 6.3)).

FFDA allocates features and data jointly among disparate views, leveraging the assumption that subsets of the data are better fit by subsets of the available features. From the standpoint of concept organization, this corresponds to different organizational schemes acting on different subsets of the available concepts (i.e. not all concepts are shared across all organizational schemes). For example, when

organizing animals by their scientific properties (e.g., *habitat, taxonomy, gestation period*) it makes sense to exclude fictional counterparts (e.g., fictional ducks such as *Donald*); however, when organizing them by their apparent physical properties (flies, quacks, has feathers), perhaps fictional animals should be included.

In the context of lexical semantics, FFDA can be motivated by considering *differential feature noise*: i.e. assumption that some features are content bearing for some subsets of data, but not for others. Identifying *when* a particular feature is spurious requires considering it in the context of the other features, and this is not a strongpoint of traditional clustering analysis.

FFDA can be defined by simply augmenting FFA with an additional random binary matrix $\mathbf{U}$ specifying which data points are included in which views:

$$\mathbf{Z}|\theta_{\mathbf{Z}} \sim \text{IBP}(\theta_{\mathbf{Z}}) \tag{6.6}$$

$$\mathbf{U}|\theta_{\mathbf{U}} \sim \text{IBP}(\theta_{\mathbf{U}}) \tag{6.7}$$

where $\mathbf{U}$ has dimension $D \times M$. Ensuring that $\mathbf{Z}$ and $\mathbf{U}$ have the same column dimensionality (number of views) can be achieved by drawing a larger matrix of dimension $(D + F) \times M$ from the IBP and partitioning it into $\mathbf{Z}$ and $\mathbf{U}$. Note that joint data and feature allocation does not significantly raise the computational complexity of the model above that of factorial feature allocation.

FFDA also requires redefining the projection operator in Equation 2.1 to operate over both the rows and columns of the data matrix, which can be realized in the obvious way. The joint operator will be written as $\mathbf{w} \circledast (\mathbf{Z}_{\cdot,m}, \mathbf{U}_{\cdot,m})$ with

the shorthand $\mathbf{w}^{\circledast(m)}$ when the feature allocation and data allocation matrices are unambiguous.

Finally, the form of the general probabilistic model must be extended to include $\mathbf{U}$ and the dependence of $\mathbf{c}$ on the data partition:

$$P(\mathbf{Z}, \mathbf{U}, \mathbf{c}|\mathbf{w}) \propto P(\mathbf{Z}, \mathbf{U}, \{\mathbf{c}^m\}, \mathbf{w}) \tag{6.8}$$

$$= P(\mathbf{Z})P(\mathbf{U}) \prod_{m=1}^{M} P(\mathbf{w}^{\circledast(m)}|\mathbf{c}^{\circledast(m)})P(\mathbf{c}^{\circledast(m)}). \tag{6.9}$$

There are several ways to account for the fact that $\mathbf{c}$ depends on $\mathbf{U}$. The simplest to extend the prior $P(\mathbf{c})$ to the entire data set, but to restrict the likelihood $P(\mathbf{w}^{\circledast(m)}|\mathbf{c}^{\circledast(m)})$ to only the data contained in the view.

6.2.4 Hierarchical Cross-Categorization

Human concept organization consists of multiple overlapping local ontologies, similar to the loose ontological structure of Wikipedia. Furthermore, each ontological system has a different set of salient properties. It would be interesting to extend MVM to model hierarchy explicitly, and compare against baselines such as *Brown clustering* (Brown et al., 1992), the nested Chinese Restaurant Process (Blei et al., 2003a) and the hierarchical Pachinko Allocation Model (Mimno et al., 2007).

Understanding the internal feature representations of concepts and how it comes to bear on conceptual organization and pragmatics is important for computational linguistic tasks that require a high degree of semantic knowledge: e.g. information retrieval, machine translation, and unsupervised semantic parsing. Fur-

thermore, feature norms have been used to understand the conceptual information people possess for the thematic roles of verbs (Ferretti, 2001).

Current fixed ontology models of conceptual organization such as WordNet cannot easily capture such phenomena (Reisinger and Paşca, 2009), although there is significant evidence for multiple organizational principles in Wikipedia categories (Chambers et al., 2010); for example people are organized by their occupation (e.g. *American politicians*), their location (e.g. *People from Queens*), or chronology (e.g. *1943 births*). Likewise, most ducks can *fly* and *quack* but only fictional ducks *appear in cartoon*s or *have nephews*; does this mean fictional ducks can be *blanched in water* and *air dried*? Accounting for the structure of such natural "tangled hierarchies," or "folksonomies," requires significantly richer models.

The cross-categorization model can be extended to *latent* hierarchical data, which requires defining a consistent model of multiple overlapping *local* categorizations within a larger hierarchical structure. Preliminary work on this model suggests that it better separates attributes according to their usage domains (Li and Shafto, 2011). Practical applications include noise-filtering for open-domain category and attribute extraction, as well as determining what terms/features are most relevant to certain query modes (classifying query intent). Evaluation of the underlying prediction models can be carried out using human annotators recruited from Mechanical Turk.

Hierarchical cross-categorization would also benefit significantly from data partitioning, as one would not expect every feature view to be relevant to *all* concepts in Wikipedia. Instead, organizational frames have a native level of generality

over which they operate, controlling what concepts are relevant to include.

6.3 Applications

6.3.1 Latent Relation Modeling

Clusterings formed from feature partitions in MVM can be viewed as a form of *implicit* relation extraction; that is, instead of relying on explicit surface patterns in text, relations between words or concepts are identified indirectly based on common syntactic patterns. For example, clusterings that divide cities by geography or clusterings partition adjectives by their polarity.

One interesting area for future work would be to characterize these latent relations in terms of their ability to suggest coherent features for relation extraction. Another possibility it to generalize the notion of selectional preference to full *frame semantics*, and evaluate how well the MVM views capture usages across different frames.

6.3.2 Latent Semantic Language Modeling

Generative models such as MVM can be used to build better priors for class-based language modeling (Brown et al., 1992). The rare n-gram results demonstrate that MVM is potentially useful for tail contexts; i.e. inferring tail probabilities from low counts.

6.3.3 Associative Anaphora Resolution

Associative anaphora[2]: are a type of bridging anaphora with the property that the anaphor and its antecedent are not coreferent, e.g.,

1. Once she saw that all **the tables**$_{(\rightsquigarrow 1)}$ were taken and **the bar**$_{(\rightsquigarrow 1)}$ was crowded, she left **the restaurant**$_{(1)}$.

2. Shares of **AAPL**$_{(2)}$ closed at $241.19. **Volatility**$_{(\leftsquigarrow 2)}$ was below the 10-day moving average.

where *tables* and *bar* in example as aspects of the *restaurant* and *volatility* in example 2 is an aspect of *AAPL* (Charolles, 1999). Resolving associative anaphora naturally requires access to richer semantic knowledge than resolving e.g. indirect anaphora, where the anaphor and its antecedent differ only by reference and can be resolved syntactically (Bunescu, 2003; Sasano and Kurohashi, 2009). The smoothed property extraction methods proposed by Reisinger and Paşca (2009) could provide a basis for performing associative anaphora resolution, hence it would be interested to do an evaluation combining it with existing coreference resolution systems (e.g. Haghighi and Klein, 2007).

Resolving associative anaphora is another domain that might potentially benefit from multi-language models. The fundamental semantic (mereological) relationships are conserved across languages, and hence resource-rich languages could be adapted for use in resource-poor languages. Note how this contrasts

[2]Also referred to as *mereological anaphora*, cf. Poesio et al. (2004).

sharply with purely syntax-level tasks, such as coreference resolution, where knowledge of the particular language structure is necessary.

6.3.4 Knowledge Acquisition

Vector-space models are commonly used in knowledge acquisition (KA), e.g. for attribute and class-instance acquisition (Lin et al., 2003; Pantel and Pennacchiotti, 2006; Van Durme and Paşca, 2008), and hence could benefit from multi-prototype and multi-view extensions, identifying relevant axes of variation along which additional high-quality data can be extracted. The current state of the art in KA ignores the downstream uses of its data, likewise, machine learning (ML) models are typically unaware of the details of the upstream KA system that generated the data. Although such functional modularity greatly simplifies system-level development, a significant amount of information is discarded that could greatly improve both systems. Several general-purpose frameworks for integrating KA and ML have been recently proposed, relying on particular model- (McCallum, 2003) or structural assumptions (Bunescu, 2008). For this project, I propose a much simpler approach: leveraging generative models of the data to predict the likelihood of specific instances or features being outliers. Such approaches are common in the statistics literature (Hoff, 2006; Verdinelli and Wasserman, 1991) but find little traction in KA.

6.3.5 Text Classification and Prediction

One straightforward way to evaluate lexical semantics models is to embed features derived from them in existing text classification and prediction problems. Comparing results to existing baselines gives a rough measure of how much additional useful semantic content is captured for that domain. Towards this end I propose evaluating the lexical semantics models on sentiment analysis (Pang et al., 2002) and predicting properties of financial text (Kogan et al., 2009).

6.3.6 Cross-Lingual Property Generation

Section 3.2.3 introduced salient property prediction as a specific application of structured lexical semantic models. Such properties are useful in downstream applications such as associative anaphora resolution (§6.3.3), but can also be evaluated on their own, e.g. comparing against human property generation norms McRae et al. (2005). Extending these models with multiple prototypes and factorial feature association is a logical next step, and would provide a coherent framework for addressing cross-language differences in concept organization.

Modeling concept structure across multiple languages simultaneously would help mitigate the noise introduced by per-language extraction idiosyncrasies and leveraging resource-rich languages to improve inference for resource-poor languages. Furthermore a large-scale comparison of concept organization norms across languages would shed light on important aspects of cross-cultural pragmatics (Wierzbicka, 1991).

6.3.7 Twitter

Twitter is a rich testbed for identifying and understanding the root causes of modern language evolution: Denotative shifts in meaning can be correlated with current events and tracked in real time. Furthermore, standardized internet-specific language features such as topical hash-tags are developing at a rapid pace, incubated primarily on Internet blogs and Twitter.

Due to its high degree of fluidity in term usage and unusually short context lengths (Phan et al., 2008; Ramage et al., 2010), traditional lexical semantics models may fail to capture interesting phenomena on Twitter. I propose applying the robust, structured models developed in this thesis to modeling the real-time lexical semantic development of Twitter hashtags. In particular, models based on DPMMs can adapt to form new clusters in real time when new data is added that does not fit well with the existing inferred structure. This ability is important since it is impossible to fix the capacity of lexical semantic models *a priori*, as new concepts (denoting current events) are constantly being added to the lexicon.

Chapter 7

Conclusion

This thesis introduced three new classes latent variable models for lexical semantics:

- The **Multi-Prototype Model** which explicitly captures ambiguity in word meaning due to homonymy by clustering individual word occurrences. The multi-prototype model is shown to perform well in cases where word senses are clearly separable, such as in generic word similarity tasks and paraphrase prediction.

- The **Tiered Clustering Model** which instruments the multi-prototype model with an additional *background* component capable of capturing word features that are shared between senses. The tiered clustering model is shown to be more suitable for modeling *polysemy*, or ambiguity with common shared structure (e.g. as in words such as *line* or *raise*) and the *selectional preference* of verbs.

- The **Multi-View Model** which divides word features up among multiple feature subsets, capturing *selective attention*, or the notion that different subsets of features are active in different word relations. The multi-view model is

shown to outperform simpler models on attribute, event, and hypernym recall as well as modeling lexical substitution for adverbs and adjectives.

The application of multi-view clustering models to distributional lexical semantics focused on its ability to (1) account for feature noise and (2) model selective attenuation by extracting coherent feature subsets that define similarity relations between words. Multi-view models are able to succinctly account for the notion that humans rely on different categorization systems for making different kinds of generalizations. These latent categorization systems underly lexical semantic phenomenon such as contextual and selectional preference, and hence may yield significant further improvements in machine translation and information retrieval. Furthermore, multi-view models can be naturally extended to model hierarchical data, inferring multiple overlapping ontologies. Such structures can be leveraged to improve, e.g., open-domain attribute and relation extraction.

Chapter 8

Bibliography

Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Paşca, M., and Soroa, A. (2009). A study on similarity and relatedness using distributional and Wordnet-based approaches. In *Proc. of NAACL-HLT-09*, pages 19–27.

Agirre, E. and Edmonds, P. (2006). *Word Sense Disambiguation: Algorithms and Applications (Text, Speech and Language Technology)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Anderson, J. (1990). *The adaptive character of thought*. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Artzi, Y. and Zettlemoyer, L. (2011). Bootstrapping semantic parsers from conversations. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 421–432. Association for Computational Linguistics.

Ashby, F. G. and Alfonso-Reese, L. A. (1995). Categorization as probability density estimation. *J. Math. Psychol.*, 39(2):216–233.

Azimi, J. and Fern, X. (2009). Adaptive cluster ensemble selection. In *IJCAI'09:*

Proceedings of the 21st international jont conference on Artifical intelligence, pages 992–997, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Baker, L. D. and McCallum, A. K. (1998). Distributional clustering of words for text classification. In *Proceedings of 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 96–103, Melbourne, Australia.

Banerjee, A., Dhillon, I., Ghosh, J., and Sra, S. (2005). Clustering on the unit hypersphere using von Mises-Fisher distributions. *Journal of Machine Learning Research*, 6:1345–1382.

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., and Etzioni, O. (2007). Open information extraction from the web. In *Proceedings of the 20th international joint conference on Artifical intelligence*.

Baroni, M. and Lenci, A. (2011). How we blessed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*. Association for Computational Linguistics.

Baroni, M., Murphy, B., Barbu, E., and Poesio, M. (2010). Strudel: A corpus-based semantic model based on properties and types. *Cognitive Science*, 34.

Baroni, M. and Zamparelli, R. (2010). Nouns are vectors, adjectives are matrices: representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1183–1193. Association for Computational Linguistics.

Barsalou, L. W. (1991). Deriving categories to achieve goals. *Psychology of Learning and Motivation-Advances in Research and Theory*, 27:1–64.

Bayer, S., Burger, J., Greiff, J., and Wellner, B. (2004). The mitre logical form generation system. In Mihalcea, R. and Edmonds, P., editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 69–72, Barcelona, Spain. Association for Computational Linguistics.

Bhattacharyya, A. (1943). On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society*, 35:99–109.

Biemann, C. and Nygaard, V. (2010). Crowdsourcing wordnet. In *Proceedings of the 5th Global WordNet conference*. ACL Data and Code Repository.

Blei, D., Griffiths, T., Jordan, M., and Tenenbaum, J. (2003a). Hierarchical topic models and the nested Chinese restaurant process. In *Proceedings of the 17th Conference on Neural Information Processing Systems (NIPS-2003)*, pages 17–24, Vancouver, British Columbia.

Blei, D., Ng, A., and Jordan, M. (2003b). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:2003.

Boster, J. S. and Johnson, J. C. (1989). Form or function: A comparison of expert and novice judgments of similarity among fish. *American Anthropologist*, 91(4):866–889.

Brody, S. and Lapata, M. (2009). Bayesian word sense induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 103–111. Association for Computational Linguistics.

Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, 18:467–479.

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

Bunescu, R. (2003). Associative anaphora resolution: A web-based approach. In *In Proceedings of the EACL2003 Workshop on the Computational Treatment of Anaphora*, pages 47–52.

Bunescu, R. (2008). Learning with probabilistic features for improved pipeline models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-08)*.

Cao, H., Jiang, D., Pei, J., He, Q., Liao, Z., Chen, E., and Li, H. (2008). Context-aware query suggestion by mining click-through and session data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM.

Chambers, A., Smyth, P., and Steyvers, M. (2010). Learning concept graphs from text with stick-breaking priors. In *Proceedings of NIPS*.

Chang, J., Boyd-Graber, J., Wang, C., Gerrish, S., and Blei, D. M. (2009a). Reading tea leaves: How humans interpret topic models. In *NIPS*.

Chang, W., Pantel, P., Popescu, A.-M., and Gabrilovich, E. (2009b). Towards intent-driven bidterm suggestion. In *Proceedings of the 18th international conference on World wide web*, WWW '09. ACM.

Charolles, M. (1999). Associative anaphora and its interpretation. In *Journal of pragmatics*, volume 31.

Chen, D. L. and Mooney, R. J. (2011). Learning to interpret natural language navigation instructions from observations. pages 859–865.

Chi, M. T., Feltovich, P. J., and Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science*, 5(2):121–152.

Clark, S. and Weir, D. (2002). Class-based probability estimation using a semantic hierarchy. *Computational Linguistics*, 28(2):187–206.

Cord, A., Ambroise, C., and Cocquerez, J.-P. (2006). Feature selection in robust clustering based on laplace mixture. *Pattern Recogn. Lett.*, 27(6):627–635.

Cruse, D. A. (1986). *Lexical Semantics*. Cambridge University Press,, Cambridge.

Cui, Y., Fern, X. Z., and Dy, J. G. (2007). Non-redundant multi-view clustering via orthogonalization. In *ICDM '07: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, pages 133–142, Washington, DC, USA. IEEE Computer Society.

Curran, J. (2004a). *From Distributional to Semantic Similarity*. PhD thesis, University of Edinburgh.

Curran, J. R. (2004b). *From Distributional to Semantic Similarity*. PhD thesis, University of Edinburgh. College of Science.

Curran, J. R. and Moens, M. (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.

Deerwester, S. C., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407.

Dhillon, I. S. and Modha, D. S. (2001). Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42:143–175.

Dinu, G. and Lapata, M. (2010). Measuring distributional similarity in context. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Emele, M. C., Dorna, M., Ldeling, A., Zinsmeister, H., and Rohrer, C. (1996). Semantic-based transfer. In *In Proceedings of the 16th International Conference on Computational Linguistics (COLING '96)*, pages 359–376.

Erk, K. (2007). A simple, similarity-based model for selectional preferences. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*. Association for Computer Linguistics.

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

Erk, K. and Padó, S. (2010). Exemplar-based models for word meaning in context. In *Proceedings of ACL*.

Fellbaum, C., editor (1998a). *WordNet: An Electronic Lexical Database and Some of its Applications*. MIT Press.

Fellbaum, C. (1998b). *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press.

Ferretti, T. (2001). Integrating verbs, situation schemas, and thematic role concepts. *Journal of Memory and Language*, 44(4):516–547.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. (2001). Placing search in context: the concept revisited. In *Proc. of the 10th international conference on World Wide Web*.

Firth, J. R. (1957). *Papers in Linguistics*. Oxford University Press.

Fountain, T. and Lapata, M. (2010). Meaning representation in natural language categories. In *Proceedings of the 32nd Annual Conference of the Cognitive Science Society*. Cognitive Science Society.

Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proc. of IJCAI-07*, pages 1606–1611.

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

Gleich, D. F. and Zhukov, L. (2004). An SVD based term suggestion and ranking system. In *ICDM '04: Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04)*, pages 391–394. IEEE Computer Society.

Goodman, N. D., Tenenbaum, J. B., Feldman, J., and Griffiths, T. L. (2008). A rational analysis of Rule-Based concept learning. *Cognitive Science*, 32(1):108–154.

Gorman, J. and Curran, J. R. (2006). Scaling distributional similarity to large corpora. In *Proc. of ACL 2006*.

Graff, D. (2003). *English Gigaword*. Linguistic Data Consortium, Philadephia.

Green, P. J. (2010). Colouring and breaking sticks: Random distributions and heterogeneous clustering. In *arXiv:1003.3988*.

Grefenstette, E., Sadrzadeh, M., Clark, S., Coecke, B., and Pulman, S. (2011). Concrete sentence spaces for compositional distributional models of meaning. *Proceedings of the 9th International Conference on Computational Semantics (IWCS11)*, pages 125–134.

Grefenstette, G. (1994). *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, Boston, Massachusetts.

Griffiths, T. and Ghahramani, Z. (2006). Infinite latent feature models and the Indian buffet process. In *Advances in Neural Information Processing Systems 18*, pages 475–482. MIT Press, Cambridge, MA.

Griffiths, T. L., Canini, K. R., Sanborn, A. N., and Navarro, D. J. (2007a). Unifying rational models of categorization via the hierarchical Dirichlet process. In *Proc. of CogSci-07*.

Griffiths, T. L., Steyvers, M., and Tenenbaum, J. B. (2007b). Topics in semantic representation. *Psychological Review*, 114:2007.

Haghighi, A. and Klein, D. (2007). Unsupervised coreference resolution in a non-parametric Bayesian model. In *Proc. ACL 2007*, pages 848–855. Association for Computational Linguistics.

Harper, K. E. (1965). Measurement of similarity between nouns. In *Proceedings of the 1965 conference on Computational linguistics*, COLING '65, pages 1–23. Association for Computational Linguistics.

Harris, Z. (1954). Distributional structure. *Word*, 10(23):146–162.

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics - Volume 2*. Association for Computational Linguistics.

Heit, E. and Rubinstein, J. (1994). Similarity and property effects in inductive reasoning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 20(2):411–422.

Herdağdelen, A. and Baroni, M. (2009). Bagpack: A general framework to represent semantic relations. In *Proc. of GEMS 2009*.

Hindle, D. and Rooth, M. (1991). Structural ambiguity and lexical relations. In *Proc. of ACL 1991*.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8).

Hoff, P. D. (2006). Model-based subspace clustering. *Bayesian Analysis*, 1(2):321–344.

Huang, C.-K., Chien, L.-F., and Oyang, Y.-J. (2003). Relevant term suggestion in interactive web search based on contextual information in query session logs. *Journal of the American Society for Information Science and Technology*, 54(7).

Jain, P., Meka, R., and Dhillon, I. S. (2008). Simultaneous unsupervised learning of disparate clusterings. In *SDM*, pages 858–869. SIAM.

Jansen, B. J., Booth, D. L., and Spink, A. (2007). Determining the user intent of web search engine queries. In *Proc. of WWW 2007*. ACM.

Jones, R., Rey, B., Madani, O., and Greiner, W. (2006). Generating query substitutions. In *Proceedings of the 15th international conference on World Wide Web*. ACM.

Joshi, A. K. and Vijay-Shanker, K. (1999). Compositional semantics with lexicalized tree-adjoining grammar (LTAG): How much underspecification is necessary? In Bunt, H. and Thijsse, E., editors, *Proc. IWCS-3*, pages 131–145.

Kilgarriff, A. (2004). How dominant is the commonest sense of a word. In *In Proceedings of Text, Speech, Dialogue*, pages 1–9. Springer-Verlag.

Kishida, K. (2005). Property of average precision and its generalization: An examination of evaluation indicator for information retrieval experiments. Technical report, NII.

Klein, D. E. and Murphy, G. L. (2001). The representation of polysemous words. *Journal of Memory and Language*, 45:259–282.

Klein, D. E. and Murphy, G. L. (2002). Paper has been my ruin: Conceptual relations of polysemous senses. *Journal of Memory and Language*, 47:548570.

Kogan, S., Levin, D., Routledge, B. R., Sagi, J. S., and Smith, N. A. (2009). Predicting risk from financial reports with regression. In *NAACL '09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 272–280, Morristown, NJ, USA. Association for Computational Linguistics.

Kruschke, J. K. (1992). ALCOVE: An exemplar-based connectionist model of category learning. *Psychological Review*, 99:22–44.

Landauer, T. and Dumais, S. (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.

Law, M. H. C., Jain, A. K., and Figueiredo, M. A. T. (2002). Feature selection in mixture-based clustering. In *Advances in Neural Information Processing Systems 15*, pages 625–632.

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

Li, D. and Shafto, P. (2011). Bayesian hierarchical cross-clustering. *Proceedings of AISTATS*, 15:443–451.

Li, L., Roth, B., and Sporleder, C. (2010). Topic models for word sense disambiguation and token-based idiom detection. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1138–1147. Association for Computational Linguistics.

Liang, P., Jordan, M. I., and Klein, D. (2011). Learning dependency-based compositional semantics. In *Proceedings of ACL*, Portland, Oregon. Association for Computational Linguistics.

Lin, D. (1998). Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2*, pages 768–774. Association for Computational Linguistics.

Lin, D. and Pantel, P. (2002). Concept discovery from text. In *Proceedings of the 19th International Conference on Computational linguistics (COLING-02)*, pages 1–7, Taipei, Taiwan.

Lin, D., Zhao, S., Qin, L., and Zhou, M. (2003). Identifying synonyms among distributionally similar words. In *Proceedings of the Interational Joint Conference on Artificial Intelligence*, pages 1492–1493. Morgan Kaufmann.

Love, B. C., Medin, D. L., and Gureckis, T. M. (2004). SUSTAIN: A network model of category learning. *Psych. Review*, 111(2):309–332.

Lowe, W. (2001). Towards a theory of semantic space. In *Proceedings of the 23rd Annual Meeting of the Cognitive Science Society*, pages 576–581.

Lund, K. and Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments & Computers*.

Ma, X., Boyd-Graber, J., Nikolova, S. S., and Cook, P. (2009). Speaking through pictures: Images vs. icons. In *ACM Conference on Computers and Accessibility*.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Mansinghka, V. K., Jonas, E., Petschulat, C., Cronin, B., Shafto, P., and Tenenbaum, J. B. (2009). Cross-categorization: A method for discovering multiple overlapping clusterings. In *Proceedings of the Nonparametric Bayes Workshop at NIPS 2009*.

Mansinghka, V. K., Kemp, C., and Tenenbaum, J. B. (2006). Structured priors for structure learning. In *Proc. UAI 2006*. AUAI Press.

McCallum, A. (2003). A note on the unification of information extraction and data mining using conditional-probability, relational models. In *In Proceedings of the IJCAI-2003 Workshop on Learning Statistical Models from Relational Data*.

McCarthy, D. and Carroll, J. (2003a). Disambiguating nouns, verbs, and adjectives using automatically acquired selectional preferences. *Computational Linguistics*, pages 639–654.

McCarthy, D. and Carroll, J. (2003b). Disambiguating nouns, verbs, and adjectives using automatically acquired selectional preferences. *Computational Linguistics*, 29(4):639–654.

McCarthy, D. and Navigli, R. (2007). SemEval-2007 task 10: English lexical substitution task. In *SemEval '07: Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 48–53, Morristown, NJ, USA. Association for Computational Linguistics.

McDonald, S. and Brew, C. (2004). A distributional model of semantic context effects in lexical processing. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

McDonald, S. and Ramscar, M. (2001). Testing the distributional hypothesis: The

influence of context on judgements of semantic similarity. In *In Proceedings of the 23rd Annual Conference of the Cognitive Science Society*.

McRae, K., Cree, G. S., Seidenberg, M. S., and McNorgan, C. (2005). Semantic feature production norms for a large set of living and nonliving things. *Behavioral Research Methods*, 37(4):547–559.

Medin, D., Ross, N., Atran, S., Cox, D., Coley, J., Proffitt, J., and Blok, S. (2006). Folkbiology of freshwater fish. *Cognition*, 99(3):237–273.

Medin, D. L. and Schaffer, M. M. (1978). Context theory of classification learning. *Psychological Review*, 85(3):207–238.

Miller, G. A. and Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28.

Mimno, D., Li, W., and McCallum, A. (2007). Mixtures of hierarchical topics with pachinko allocation. In *ICML*.

Mitchell, J. and Lapata, M. (2008). Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244, Columbus, Ohio. Association for Computational Linguistics.

Montague, R. (1973). The proper treatment of quantication in ordinary English. In Hintikka, J., Moravcsik, J., and Suppes, P., editors, *Approaches to Natural Language*, pages 221–242.

Murphy, G. L. (2002). *The Big Book of Concepts*. The MIT Press.

Neal, R. M. (2000). Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265.

Nelson, L. J. and Miller, D. T. (1995). The distinctiveness effect in social categorization: You are what makes you unusual. *Psychological Science*, 6:246–249.

Nida, E. A. (1975). *Componential Analysis of Meaning: An Introduction to Semantic Structures*. Mouton, The Hague.

Niu, D., Dy, J. G., and Jordan, M. I. (2010). Multiple non-redundant spectral clustering views. In Fürnkranz, J. and Joachims, T., editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 831–838.

Nosofsky, R. M. (1986). Attention, similarity and the identication categorization relationship. *Journal of Experimental Psychology: General*, 115:39–57.

Nosofsky, R. M., Palmeri, T. J., and McKinley, S. C. (1994). Rule-plus-exception model of classification learning. *Psychological Review*, 101(1):53–79.

Ó Séaghdha, D. (2010). Latent variable models of selectional preference. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL-10)*, Uppsala, Sweden.

Ó Séaghdha, D. and Korhonen, A. (2011). Probabilistic models of similarity in syntactic context. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP-11)*, Edinburgh, UK.

Paşca, M., Lin, D., Bigham, J., Lifchits, A., and Jain, A. (2006). Names and similarities on the web: fact extraction in the fast lane. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

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

Padó, S., Padó, U., and Erk, K. (2007). Flexible, corpus-based modelling of human plausibility judgements. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 400–409, Prague, Czech Republic. Association for Computational Linguistics.

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

Palmer, F. R. (1976). *Semantics*. Cambridge University Press.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86.

Pantel, P., Bhagat, R., Chklovski, T., and Hovy, E. (2007). ISP: Learning inferential selectional preferences. In *In Proceedings of NAACL 2007*.

Pantel, P. and Lin, D. (2002). Discovering word senses from text. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 613–619, New York, NY, USA. ACM.

Pantel, P. and Pennacchiotti, M. (2006). Espresso: leveraging generic patterns for automatically harvesting semantic relations. In *ACL '06: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 113–120, Morristown, NJ, USA. Association for Computational Linguistics.

Pantel, P. A. (2003). *Clustering by committee*. PhD thesis, Edmonton, Alta., Canada.

Pedersen, T. and Kulkarni, A. (2006). Automatic cluster stopping with criterion functions and the gap statistic. In *Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 276–279, Morristown, NJ, USA. Association for Computational Linguistics.

Pennacchiotti, M., De Cao, D., Basili, R., Croce, D., and Roth, M. (2008). Automatic induction of framenet lexical units. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Pereira, F., Tishby, N., and Lee, L. (1993). Distributional clustering of English words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics (ACL-93)*, pages 183–190, Columbus, Ohio.

Phan, X.-H., Nguyen, L.-M., and Horiguchi, S. (2008). Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 91–100, New York, NY, USA. ACM.

Pitman, J. (1995). Exchangeable and partially exchangeable random partitions. *Probab. Theory Related Fields*, 102(2):145–158.

Pitman, J. and Yor, M. (1997). The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *Annals of Probability*, 25(2):855–900.

Poesio, M., Mehta, R., Maroudas, A., and Hitzeman, J. (2004). Learning to resolve bridging references. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 143, Morristown, NJ, USA. Association for Computational Linguistics.

Poon, H. and Domingos, P. (2009). Unsupervised semantic parsing. In *Proc. of EMNLP 2009*. Association for Computational Linguistics.

Posner, M. I. and Keele, S. W. (1968). On the genesis of abstract ideas. *Journal of Experimental Psychology*, 77(3):353–363.

Pothos, E. and Chater, N. (2002). A simplicity principle in unsupervised human categorization. *Cognitive Science*, 26(3):303–343.

Pothos, E. M. M. and Close, J. (2007). One or two dimensions in spontaneous classification: A simplicity approach. *Cognition*.

Pustejovsky, J. (1995). The generative lexicon. *Computational Linguistics*, 17.

Ramage, D., Dumais, S., and Liebling, D. (2010). Characterizing microblogs with topic models. In *ICWSM*.

Ramage, D., Hall, D., Nallapati, R., and Manning, C. D. (2009a). Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 248–256, Singapore. Association for Computational Linguistics.

Ramage, D., Rafferty, A. N., and Manning, C. D. (2009b). Random walks for text semantic similarity. In *Proc. of the 2009 Workshop on Graph-based Methods for Natural Language Processing (TextGraphs-4)*, pages 23–31.

Rapp, R. (2003). Word sense discovery based on sense descriptor dissimilarity. In *Proceedings of the Ninth Machine Translation Summit*.

Rasmussen, C. E. (2000). The infinite Gaussian mixture model. In *Advances in Neural Information Processing Systems*, pages 554–560. MIT Press.

Reisinger, J. and Mooney, R. (2010). Multi-prototype vector-space models of word meaning. In *Proc. of NAACL 2010*. Association for Computational Linguistics.

Reisinger, J. and Paşca, M. (2009). Latent variable models of concept-attribute attachment. In *Proc. of ACL 2009*, pages 620–628. Association for Computational Linguistics.

Reisinger, J. and Pasca, M. (2011). Fine-grained class label markup of search queries. The Association for Computer Linguistics.

Resnik, P. (1997). Selectional preference and sense disambiguation. In *Proceedings of ACL SIGLEX Workshop on Tagging Text with Lexical Semantics*, pages 52–57, Washington, D.C. ACL.

Ritter, A., Mausam, and Etzioni, O. (2010). A latent Dirichlet allocation method for selectional preferences. In *In Proceedings of ACL 2010*.

Rooth, M., Riezler, S., Prescher, D., Carroll, G., and Beil, F. (1999). Inducing a semantically annotated lexicon via EM-based clustering. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 104–111, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ross, B. H. and Murphy, G. L. (1999). Food for thought: Cross-classification and category organization in a complex real-world domain. *Cognitive Psychology*, 38:495–553.

Rudolph, S. and Giesbrecht, E. (2010). Compositional matrix-space models of language. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 907–916, Uppsala, Sweden. Association for Computational Linguistics.

Salton, G., Wong, A., and Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Sanborn, A. N., Griffiths, T. L., and Navarro, D. J. (2006). A more rational model of categorization. In *Proceedings of the 28th Annual Conference of the Cognitive Science Society*.

Sanderson, M. (1994). Word sense disambiguation and information retrieval. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 142–151, New York, NY, USA. Springer-Verlag New York, Inc.

Sasano, R. and Kurohashi, S. (2009). A probabilistic model for associative anaphora resolution. In *EMNLP '09: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1455–1464, Morristown, NJ, USA. Association for Computational Linguistics.

Schütze, H. (1998a). Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

Schütze, H. (1998b). Automatic word sense discrimination. *Comput. Linguist.*, 24(1):97–123.

Shafto, P. and Coley, J. D. (2003). Development of categorization and reasoning in the natural world: Novices to experts, naive similarity to ecological knowledge. *Journal of Experimental Psychology: Learning, Memory, and Cognition*.

Shafto, P., Kemp, C., Mansinghka, V., Gordon, M., and Tenenbaum, J. B. (2006). Learning cross-cutting systems of categories. In *Proc. CogSci 2006*.

Shan, H. and Banerjee, A. (2010). Residual Bayesian co-clustering for matrix approximation. In *SIAM International Conference on Data Mining (SDM) 2010*.

Shepard, R. N., Hovland, C. L., and Jenkins, H. M. (1961). Learning and memorization of classifications. *Psychological Monographs*, 57.

Smith, E. R., Fazio, R. H., and Cejka, M. A. (1996). Accessible attitudes influence categorization of multiply categorizable objects. *Journal of personality and social psychology*, 71(5):888–98.

Smith, N. A. and Shafto, P. (2011). The role of cross-cutting systems of categories in category-based induction. In *Proceedings of the 33rd Annual conference of the Cognitive Science Society*.

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46:159–216.

Snow, R., Jurafsky, D., and Ng, A. (2006). Semantic taxonomy induction from heterogenous evidence. In *Proc. of ACL 2006*.

Snow, R., O'Connor, Jurafsky, D., and Ng, A. (2008). Cheap and fast—but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of Empirical Methods in Natural Language Processing*.

Spärck Jones, K. (1964). *Synonymy and Semantic Classification*. PhD thesis, University of Cambridge.

Strehl, A. and Ghosh, J. (2003). Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617.

Tatu, M. and Moldovan, D. (2005). A semantic approach to recognizing textual entailment. In *Proc. of HLT-EMNLP 2005*. Association for Computational Linguistics.

Teh, Y. W., Daumé III, H., and Roy, D. (2007). Bayesian agglomerative clustering with coalescents. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, Vancouver, Canada.

Thater, S., Fürstenau, H., and Pinkal, M. (2010). Contextualizing semantic representations using syntactically enriched vector models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Thater, S., Fürstenau, H., and Pinkal, M. (2011). Word meaning in context: A simple and effective vector model. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1134–1143.

Tokunaga, K., Kazama, J., and Torisawa, K. (2005). Automatic discovery of attribute words from Web documents. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP-05)*, pages 106–118, Jeju Island, Korea.

Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. In *Proc. of the ACL*.

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

Turney, P. D. and Pantel, P. (2010). From frequency to meaning: vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1).

Tversky, A. and Gati, I. (1982). Similarity, separability, and the triangle inequality. *Psychological Review*, 89(2):123–154.

Van de Cruys, T., Poibeau, T., and Korhonen, A. (2011). Latent vector weighting for word meaning in context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1012–1022. Association for Computational Linguistics.

Van Durme, B. and Paşca, M. (2008). Finding cars, goddesses and enzymes: Parametrizable acquisition of labeled instances for open-domain information extraction. In *Proc. of AAAI 2008*.

Verdinelli, I. and Wasserman, L. (1991). Bayesian analysis of outlier problems using the Gibbs sampler. *Statistics and Computing*, 1(2).

Voorspoels, W., Vanpaemel, W., and Storms, G. (2009). The role of extensional information in conceptual combination. In *Proceedings of the 31th Annual Conference of the Cognitive Science Society*.

Vyas, V., Pantel, P., and Crestan, E. (2009). Helping editors choose better seed sets for entity set expansion. In *Proceedings of the 18th ACM conference on Information and knowledge management*.

Wen, J.-R., Nie, J.-Y., and Zhang, H.-J. (2001). Clustering user queries of a search engine. In *Proceedings of the 10th international conference on World Wide Web*. ACM.

Wierzbicka, A. (1991). *Cross-cultural pragmatics : The semantics of human interaction*. Mouton de Gruyter, Berlin ; New York.

Williamson, S., Wang, C., Heller, K. A., and Blei, D. M. (2010). The IBP-compound Dirichlet process and its application to focused topic modeling. In *Proceedings of the 27th International Conference on Machine Learning*.

Xue, N., Chen, J., and Palmer, M. (2006). Aligning features with sense distinction dimensions. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 921–928, Morristown, NJ, USA. Association for Computational Linguistics.

Yao, X. and Durme, B. V. (2011). Nonparametric Bayesian word sense induction. In *Proceedings of TextGraphs-6: Graph-based Methods for Natural Language Processing*, pages 10–14. Association for Computational Linguistics.

Zarate, M. A. and Smith, E. R. (1990). Person categorization and stereotyping. *Social Cognition*, 8(2):161–185.

Zettlemoyer, L. S. and Collins, M. (2005). Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of 21st Conference on Uncertainty in Artificial Intelligence (UAI-2005)*, Edinburgh, Scotland.

Zipf, G. (1935). *The Psycho-Biology of Language*. Houghton Mifflin, Boston, MA.