Lexicography as Feature Engineering: automatic discovery of distinguishing semantic factors for synonyms

Master’s Thesis

Presented to

The Faculty of the Graduate School of Arts and Sciences
Brandeis University
Graduate Program in Computational Linguistics
James Pustejovsky, Advisor

In Partial Fulfillment
of the Requirements for the Degree

Master of Arts
in
Computational Linguistics

by
Orion Montoya

August, 2017
Copyright © 2017

by Orion Montoya
Acknowledgments

I might not have picked up this line of research or returned to grad school at all if Ilan Kernerman had not asked me to review Patrick Hanks’s *Lexical Analysis: Norms and Exploitations*, so for that invitation I am grateful to both of them. When my career wandered away from practical lexicography, I had an inkling that someone else would independently begin pursuing the theoretical problems I found most interesting. As far as I could see two years ago, this area of research remained pretty quiet, so I came see if I could make a contribution.

Thanks of course to James Pustejovsky and Jaco, whose encouragement and fluffiness helped me see that the track I am on is less-traveled not because it does not lead somewhere useful, but because it is both foggy and difficult — and James pointed me to the brightest lights within the fog.
Abstract

Lexicography as Feature Engineering: automatic discovery of distinguishing semantic factors for synonyms

A thesis presented to the Graduate Program in Computational Linguistics
Graduate School of Arts and Sciences
Brandeis University
Waltham, Massachusetts
By Orion Montoya

Many different techniques can group synonymous words together, but it is much harder to untangle the nuance and emphasis borne by individual members of a synonym set. The distributional hypothesis holds that you may understand a word by the contexts in which it appears. This thesis applies a sort of transposition of the distributional hypothesis to groups of synonyms. I postulate that, by aggregating the differences among the contexts of similar words, one may discover the range of semantic factors borne by a synonym group as a whole, as well as the implications of choosing an individual member of a synset. By applying an analytical framework along these lines to extract semantic factors for set of synonyms, I initially find factors that are of a different sort than the factors used in most decompositional analyses, and see promise in developing the approach further with varying methods of feature extraction.
Contents

Abstract \hspace{2cm} iv

1 Loomings \hspace{2cm} 1
  1.1 Introduction \hspace{2cm} 1
  1.2 Elaborating the hypothesis \hspace{2cm} 6

2 Background \hspace{2cm} 8
  2.1 Ontology and Lexicography \hspace{2cm} 8
  2.2 Factorization & Decomposition \hspace{2cm} 16
  2.3 Pairwise comparison is practicable with current tools; groupwise comparisons are not. \hspace{2cm} 18

3 Experimental Setup \hspace{2cm} 24

4 Results \hspace{2cm} 32
  4.1 Adverbial Modifiers Only \hspace{2cm} 32
  4.2 Usage Patterns \hspace{2cm} 36
  4.3 Full-wordsketch features \hspace{2cm} 39

5 Conclusion \hspace{2cm} 44
  5.1 Observations \hspace{2cm} 44
  5.2 Future Work \hspace{2cm} 45
  5.3 Finally \hspace{2cm} 51
Chapter 1

Loomings

1.1 Introduction

1.1.1 The Problem

Many dozens of ontologies and thesauruses can tell you words that have similar meanings and how they group together. Distributional statistics can also find interchangeable words and grade their degree of interchangeability in a reduced-dimensionality vector space. Beyond a certain level of sense division, however, no pair of synonyms is completely equivalent or interchangeable. Neither ontological nor distributional resources are built to enable systematic characterization of specific nuances of lexical units within their similarity groupings.

Thesauruses for humans typically, and notoriously, offload the burden of semantic nuance onto their users. Human users choose synonyms based either on their pre-existing knowledge of meaning which is merely refreshed by the thesaurus, or on dictionary lookups of words whose meanings they cannot recall with confidence. In either case, the thesaurus leans heavily on the user’s established linguistic sensitivity and cultural competence. In computational applications, such competence cannot be presumed, and existing lexical resources have not
CHAPTER 1. LOOMINGS

been used as a dependable disambiguator of synonyms.

If no computational resource exists that can give an explanatory account of the differences between members of a set of synonyms, what would be necessary to make one? If a dictionary were written in a particularly systematic way, such that its meaning descriptions were reliably comparable between arbitrary pairs of words, it might serve as a valuable supplement to thesauruses for addressing questions of intensional nuance. But existing resources can offer little more than a very crudely educated guess. Occasionally, hand-built resources have attempted such explications at a small scale, but to date never on the scale of the whole lexicon. This is a significant gap, which limits the possibilities of computing meaning.

Beyond making more useful thesauruses, semantically sensitive lexical selection is a requirement of effective natural language generation (NLG), and finer-grained knowledge of lexical semantic factors would also yield copious benefits for natural language understanding (NLU). An ideal system should be able to generate or understand ‘stare’ when given ‘look intently’, and ‘look briefly’ when given ‘glance’, without stumbling into markedly odd alternatives like ‘stare briefly’. Raw or smoothed frequencies of collocations can get some of the way there, but human judgment is likely necessary to judge the difference between combinations that are rare because they describe less-common occurrences in reality, and those that are unattested because they are semantically bizarre.

The biggest obstacle to recording human judgments of synonym distinctions is that evaluating synonyms as a group requires a large \(O(N^2)\) number of pairwise comparisons, but that these comparisons become confusing and hard to manage even in pretty small \(N\). The best corpus tools (e.g. SketchEngine [Kilgarriff et al., 2014]) are organized to show the behavior of individual or, at most, paired lexical units. The primary goal of the present thesis is enable comparison of synonyms in groups, for both human and computational applications.
CHAPTER 1. LOOMINGS

1.1.2 Desiderata

In the present state of computational linguistics and lexicography, a number of qualities are both desirable and achievable in a lexical resource. It goes almost without saying that a lexical resource should be based on a large corpus of the language it claims to describe. It should be understood that the meanings in a lexical resource reflect only the meanings in the corpus that underlies it (although most certainly not every meaning in the corpus). Other quite achievable attributes, however, are not yet common.

Dictionaries commonly make definitional claims about semantic factors of lexical items without grounding them in any particular evidence. It is desirable and achievable, but not yet common, for a lexical resource to have all of its human-made judgments connected explicitly to the evidence in the corpus that forms the basis of these judgments. The Pattern Dictionary of English Verbs (PDEV) [Hanks and Pustejovsky, 2005] uses Corpus Pattern Analysis to map between syntactic patterns, semantic-role arguments and enumerated word senses. PDEV analysis is performed within the same corpus (the British National Corpus (BNC)) from which the sense inventory (the New Oxford Dictionary of English, [Pearsall and Hanks, 1998]) was derived, so its mappings have the same evidentiary basis — but PDEV is a retrofit: re-analyzing samples of the same data to record a new layer of judgments. FrameNet [Baker et al., 1998] similarly co-develops annotated texts with its inventory of semantic frames. Both of these show when and where an item in their inventory is attested in the corpus, but only for individual, specific high-level aspects of argument structure.

It is desirable, and probably achievable, for meaning descriptions to be comparable. The overwhelming majority of meaning descriptions in computational use are those that come from WordNet, where members of a synset are effectively treated as perfect equivalents. Techniques for comparing synsets are overwhelmingly likely to be based on the Lesk algorithm [Lesk, 1986]. Lesk similarity is highly sensitive to the specific phrasing of definitions,
which are themselves typically written with semantic density and formal constraints comparable to haiku. Definitions in native-speaker dictionaries lean heavily on shared cultural knowledge. In good learners’ dictionaries the cultural baggage is substantially lighter, with lexicographers making an effort to provide context to ground a learner in the frames of reference that motivate the use of a word. But even in those cases the explicatory emphasis is on specific expected cultural quirks of the target language, and not on a ground-up progression from first principles.

It is desirable and probably achievable for aspects of semantic description to be extracted from corpus evidence in, at worst, a semi-supervised fashion, but this is not widespread. One step in this direction is ‘tickbox lexicography’ [Kilgarriff et al., 2010], wherein an analyst may conveniently cherry-pick collocations and examples to illustrate a lexical unit under investigation. I am not aware of other recent work that attempts this at the word level, although in the course of this research I have seen some clear affinities with topic modeling (for better and for worse). Work that has grown out of [Stefanowitsch and Gries, 2003], such as [Desagulier, 2012], also has some affinity with this thesis’s techniques, but as far as I can tell still aims at smaller-scale comparisons with more syntactic than semantic emphasis. Furthering the realization of this achievable desideratum — semi-supervised extraction of semantic descriptions — is the focus of this thesis.

1.1.3 The hypothesis

One formulation of the distributional hypothesis, generally traced to both [Firth, 1957], and [Harris, 1954], holds that you may understand words by the contexts in which they appear. In this paper I transpose the distributional hypothesis to look for distinctions and commonalities within the contexts of groups of words whose meanings and contexts are generally similar. It is widely accepted that a word’s typical lexical-syntactic environments
contain actionable clues to its meaning potentials. The assumption underlying my hypothesis is that collocating terms, especially in dependency relations to a head word, state explicitly when certain semantic features of their heads must be understood, and that on the scale of a large corpus, these statements accrete to paint a portrait of the words. I postulate that, by aggregating the differences among the contexts of similar words, we may discover both the range of semantic factors borne by a synonym group as a whole, and the implications of choosing an individual member of a synset.

On its own, the hypothesis that it is useful to compare contexts of words is not new; it is a foundational principle of corpus lexicography in general, and since at least the 1970s is has been the basis of Apresjan’s [2000] manual process for explicating synonyms. The new contribution in my approach is that it begins with unsupervised clustering of synonym sets based on features of each word’s contexts, then extracts the most salient features of those clusters to see which meaning elements are normally (in a Hanksian sense [Hanks, 2013]) emphasized for some synonyms and not others. This pre-clustering step may allow more focused and efficient comparisons of relevant factors than other approaches.

I seek to show the potential of this framework, and to demonstrate that it is worth developing further. The results I present do not come close to solving all of the problems of semantic description, or of extracting salient semantic factors in an unsupervised manner. But they do show that the approach — with selection of clustering algorithms, cluster labeling methods, and especially feature engineering in the control of the lexical analyst — has the capacity to accelerate groupwise corpus analysis of synonym sets.

The output of the automatic clustering and labeling is somewhat useful on its own, but even more useful as a starting point for further investigation and iteration. I would like to show that this framework, with its theoretical assumptions and practical outcomes, can enable more nuanced semantic description than is practicable with current techniques, which
in turn makes possible applications that are not conceivable with current resources.

1.2 Elaborating the hypothesis

Wittgenstein, *Philosophical Investigations* §79: *(The fluctuation of scientific definitions: what today counts as an observed concomitant of phenomenon A will tomorrow be used to define “A”.)

**Hypothesis 0a**: The semantic range of a word’s collocations may be used as a proxy for the semantic range of the word itself. Words do not habitually appear in syntactic dependencies with words that are irrelevant to their semantics: instead, the normal meanings of a word are colored by the words they typically appear with. This association bears out through all of a word’s dependency relations. Subjects: executions are normally carried out by organizations, murders by individuals. [Hanks, 2013, p. 245] cites a BNC story where ‘boys . . . beheaded a few pigeons’ to illustrate that behead is being exploited, because it ‘normally denotes a form of judicial capital punishment, not merely an act of cutting off the head’. Objects: one may cut cake or cheese, but not rice. Adverbs, as we will see in abundance: one may ‘angrily demand’, one might ‘angrily suggest’, but it would be unusual to ‘angrily encourage’.

**Hypothesis 0b**: Near-synonyms are commonly distinguished from one another by modification: synonymous verbs have their differences emphasized by adverbs, nouns by adjectives, etc. This is not the only role that modifiers play, of course, but it may be frequent enough to be significant. Any writing that is less than perfectly economical will contain partially pleonastic descriptors to emphasize the semantics of carefully-chosen words, or to enrich the semantics of general words.

**Hypothesis 1**: If distributed dependency relations are a proxy for a word’s semantics as in 0a, and near-synonyms are distinguished in prose by their modifiers as in 0b, then we
CHAPTER 1. LOOMINGS

should be able to find and characterize these distinguishing modifiers in a dependency-parsed corpus.

1.2.1 Testing the hypothesis

We develop a technique to automatically extract semantic features from a corpus by counting salient differences in distributional contexts between synonyms. Broadly speaking this may be seen as a new systematization of Apresjan’s methods for creating comprehensive descriptions of the semantic similarities and differences between members of a group of synonyms. In the short term, at least, such analysis cannot be done completely automatically with our methods; it must be done semi-manually. Thus the emphasis on feature engineering. The work of an analyst in this framework is to:

- Semi-manually gather sets of synonyms to analyze: combining ontological similarity, distributional similarity, and judgment;

- Engineer features to characterize the salient contextual attributes of the meaningful contexts of the words in use;

- Automatically cluster the synonyms on these features and analyze the clusters in a semantically sensitive way;

- Repeat the feature engineering and clustering until the synonymous words have been lumped (or split) and labeled to the lexicographer’s satisfaction dd

If this works, there are immediate possibilities for lexicographical processes, and also future opportunities for more nuanced computational treatments of lexical semantics.
Chapter 2

Background

This thesis touches on lexicography and ontology; on semantic factorization and decomposition; and on vector-space distributional semantics. In this chapter I situate my work within each area.

2.1 Ontology and Lexicography

2.1.1 (Onomasiological) Ontology

The organization of concepts into hierarchies has a long history. Since the Enlightenment, Francis Bacon and Diderot and d’Alembert both sorted human knowledge into tree structures [Bacon, 1605], [Diderot and d’Alembert, 1751]. John Wilkins had a specifically linguistic motivation to classified concepts with as a preliminary to his seminal conlang [Wilkins, 1668], shown in figure 2.1. But Peter Mark Roget is the one whose organization of words by meanings [Roget, 1852] became an enduring standard.

Aside from their inherently quixotic motivation, the striking commonality among ontologies is that each one is different. As Wittgenstein notes, “...how we group words into kinds
Figure 2.1: John Wilkins’s outline of his “Scheme or Analysis of all the Genus’s”, [Wilkins, 1668]
It is unwise to use an ontology as a tool without understanding the aims of its classification system and the inclinations of its creators. Roget’s *Thesaurus* has been an asset to computational linguists since the 1950s (see a good account in [Jarmasz, 2003]) but did not exist in a really tractable version until well after the advent and supremacy of WordNet.

Kurt Baldinger [Baldinger, 1980] discusses the different ways that ‘language divides up the world’, and shows several diagrams from Coseriu in which the differences between closely-related concepts are expressed according to various schemata: two different shapes of trees (figs. 2.2, 2.3), and a matrix of features (fig. 2.4).

These schemata are the product of specific semantic analysis of a narrow conceptual space for purposes of semantic theory, and to my knowledge have never been attempted at the scale of a whole lexicon. Some of the judgments seen in these diagrams may, with effort, be incidentally recoverable from resources like WordNet/GermaNet/EuroWordNet,
CHAPTER 2. BACKGROUND

Figure 2.3: Baldinger’s reproduction of A. Greimas’s graphic chart of the conceptual system of German ‘Schall’. [Baldinger, 1980][p. 106]

Figure 2.4: Baldinger’s reproduction of Pottier’s graphic analysis of the conceptual system of German ‘Schall’. [Baldinger, 1980][p. 107]
but expressing onomasiological distinctions is not directly supported by the schemata of Word Nets.

Lexicographically-minded enrichment of lexical resources with explicit contrasts between similar words has been occasionally toyed with since the latter half of the 20th century, but, again, never at lexicon scale. Zgusta [1971] reproduces a schematic graph (Figure 2.5) of a set of humor-related words from [Schmidt-Hidding, 1963]. The chart maps the relation between various kinds of wit and humor, on the primary axes of “love ↔ hate” and “wit ↔ farce.” Zgusta warns that “it would be illusory to think or even postulate that the lexicographer could or should produce studies like [this one] as a by-product of his own, lexicographic work, or in the way of preparation for his work: he will never have time for that, at least not in the usual cases.”

In 1979, on the other hand, Apresjan writes that “a dictionary of synonyms must contain a full, explicit and non-redundant description of the similarities and differences between [synonyms] in their co-occurrence constraints—both lexico-semantic and syntactic” (reprinted in
CHAPTER 2. BACKGROUND

[Apresjan, 2000, p. 5]. Apresjan’s systematic approach amounts to taking groups of synonyms and characterizing all of their contrasting dimensions as a preliminary to organizing them in a thesaurus: the same task that Zgusta warns there is can never be enough time to do.

Indeed what Apresjan proposes is even larger than what Zgusta warns against: “The complete semantic description of a word, which is of primary importance in a dictionary of synonyms, is broader than its meaning, and includes, in addition, information about its presuppositions, the semantic associations, or connotations, that it evokes, its logical stress, and a number of other matters of less concern here” [2000, p. 19].

Apresjan’s Anglo-Russian Dictionary of Synonyms [Apresjan and Botiakova, 1979] is the glorious product of this proposal — a set of studies of several hundred groups of synonymous or plesionymous words, contrasting them by their various implications, selectional preferences, etc. Apresjan guides much of what we do below. For the moment the significance of Apresjan’s dictionary is on two hands. On one hand, it exists: showing that systematically distinguishing synonymous groups is undeniably possible, even in a pre-corpus age. On the other, it does require a lot of work, takes up a lot of space, and is described in pure prose, not in a computationally tractable form. Efficient allocation of space was a persistent concern in the print era. For the Anglo-Russian Dictionary of Synonyms there was clearly a minimum size of synonym group that was worthy of consideration, a maximum generality of a synonym group to allow it to be entered at a single alphabetical entry point, and therefore a natural tendency to explicate moderate-sized groups that exhibit high semantic commonality. Comparison within distant groups is left as an exercise for the reader.

The Dictionnaire Explicatif et Combinatoire du Français Contemporain [Mel’cuk et al., 1984] is perhaps the longest-running concerted attempt at semantic descriptions in schemata that might allow for comparison. Description in terms of lexical functions has some really
exciting possibilities: in fig. 2.6, an excerpt from DECFC’s entry for **malade**, we can see both register markers for the **Magn**, and graded near-synonyms at the **AntiMagn** – **souffrant** and **indisposé** being alternative near-synonyms that express something near in meaning to **malade**, but with weaker import. The insurmountable obstacle to using DECFC entries as the basis for broad onomasiological comparisons is that there just aren’t very many entries to work from: covering only a couple hundred words since first publishing in 1984, the project has a long way to go.

Wierzbicka and Goddard’s work in Natural Semantic Metalanguage [Wierzbicka, 1985], [Goddard and Wierzbicka, 1994], [Goddard and Wierzbicka, 2013] has yielded a dense trickle of thorough, and potentially very computationally useful, descriptions of individual words, but their pace, too, is Sisyphean. And none of the significant approaches to date have been corpus-driven. The best, most blue-sky thing to be hoped for is a way of computationally accelerating the production of resources like DECFC and NSM definitions, with a foundation in statistical patterns found in a corpus. This thesis is a step towards such a resource, though
the ultimate output may take a very different shape than the ones we have seen to date.

WordNet [Miller, 1995], it may go without saying, is the most commonly-used ontological resource in NLP. It holds this position due to its early implementation, comprehensiveness, multiple dimensions of semantic relations, and usable API. Its most striking contrast from lexicographical or onomasiological ontology is deep in its roots: from its inception, WordNet was intended as a psycholinguistic resource structured around particular psycholinguistic theories of lexical access [Miller et al., 1990]. Its adaptation to other applications is therefore something of an accident of history. Its flaws for the purposes it has been put to are evident to anyone who engages with it; the critique given in [Hanks and Pustejovksy, 2005] may suffice to account for its generally poor fit for ontological and lexical-semantic purposes. It is necessary to mention it, however, as an example of the consequences of mis-fit between the aims of a classification system and of the people using it. WordNet’s ‘psycholexicological’ motivation can explain many of its puzzlements — such as listing nearly-identical senses in multiple places in the hierarchy — but such explanation makes WordNet no less confounding for semantic applications. Wordnet is an artifact of cognitive science, retrofitted to computational linguistic applications because it was the most comprehensive thing available.

With the ‘aims of the classification’ appropriately managed, ontological arrangements work well for managing meaning in a structured way — right up to the point of distinguishing synonyms, where they become useless. A leaf of either WordNet or Roget amounts to a list of words which, for the ontology’s purposes, are equivalent. But it is exactly at this point that lexical choice becomes interesting, and the finer distinctions become relevant.

2.1.2 Lexicography

The transformation of ontological hierarchies into semantically-descriptive thesauruses is somewhat under-studied in both lexicography and metalexicography. Roget’s ontology was
early augmented with an alphabetical index, and many resources are now purely alphabetical, using a combination of cross-reference and repetition to meet users’ information needs with minimal page-flipping (all common thesauruses, save WordNet, still having been born in a print mentality). In almost all cases, regardless of how one gets there, at the end one finds lists of words without explicit differentiation.

Some thesauruses in the second half of the 20th century began to include comparative paragraphs at synonym groups that were particularly tricky (or irresistibly gradient). Apresjan’s thesaurus is the only one I know to systematically compare meanings for every listed synonym group. Many metalexicographic authors share Zgusta’s hopelessness at the notion of putting every content word in an ontology into a comparative structure.

Defining with genus and differentiae, we take the same path as we do with Roget or Wordnet, and we end up marooned in a similar place. Generative Lexicon allows us to compute meanings and understand Formal, Constitutive, Agentive and Telic roles. Ontologies seem to focus especially on the Formal and Constitutive aspects. Telicity — or nuances of telicity — tends to be one of the things that distinguish near synonyms; Agentive perhaps less so — but perhaps we will see about that.

2.2 Factorization & Decomposition

In the past century small, discrete subcomponents of meaning have been approached in broadly comparable ways by different linguistic disciplines — generative semantics, lexicography, ontology — called by different names: components ([Goodenough, 1956], [Wierzbicka, 1985], [Apresjan, 2000]), semantic features [Katz and Fodor, 1963], [Zgusta, 1971], lexical factors [Joshi, 1972], [Pustejovsky and Joshi, 2017], lexical functions [Mel’čuk, 1996], [Mel’čuk, 1998], among others. These approaches have their differences of theoretical framing, practical application, and combinatorial syntax, but a common factor is that they tend
to involve the decoration of lower-level conceptual primitives with distinctive meaning factors borne by individual lexical items.

Semantic factors such as these also form the skeleton of ontologies like Roget and WordNet: each parent node in the hierarchy can be understood to encode an atomic common property of its children. Theoretically, this property of ontologies could be used to bootstrap a semantic factorization. But keeping in mind Wittgenstein’s dictum about the aims of a classification, it would be dangerous to do this mechanistically based on a resource that did not have factorization as an aim.

Collocational and distributional phenomena in a large textual corpus present an obvious empirical path toward discovering the differences of emphasis that follow from the choice of one synonym over another. Building on [Joshi, 1972], Pustejovsky and Joshi [2017] revisited lexical factorization (of verbs of seeing) with a corpus-driven, empirical approach, testing and measuring likelihood that a semantic component of a lexical item will be overtly expressed in composition. That work starts from pre-factorized meaning components, singling out collocating modifiers that independently express these components. The present paper contains experiments going in the opposite direction: from syntactically-expressed verbal arguments, in comparison with the distribution of other plesionymous words, can we discover a usefully distinguishing subset of the components of a given word’s meaning?

2.2.1 Beyond antonymy and hyperonymy

Other interesting work has sought to distinguish synonyms from their distributionally-similar antonyms [Mohammad et al., 2013], or to determine which neighbors in a vector space denote a concept more general than the others [Kotlerman et al., 2010]. These are important questions to answer, but they still only scratch at the surface of lexical meaning. Pug and labradoodle are both hyponyms of dog, but when one or the other walks into a scene, they
CHAPTER 2. BACKGROUND

bring very different intensional baggage with them. I am looking for factors of meaning that reflect words’ habitual intensional associations.

2.3 Pairwise comparison is practicable with current tools; groupwise comparisons are not.

The state-of-the-art place to look for associations of individual words is through collocation in a corpus. The Sketch Engine [Kilgarriff et al., 2014] counts and normalizes collocation distributions to provide a succinct snapshots of individual lemmas, conveniently sorted into groups by the grammatical relation between the collocate and the target lemma. The grammatical relations may be those identified by a dependency parser, but more commonly they are identified by hand-written grammars configured for each corpus’s tag set.

Figures 2.7 and 2.8 show the word sketches for convince as identified by two different taggers. The comparison is informative of the fact that, despite the abundance of training instances offered by a large corpus, feature selection is still a major determinant of the kinds of results possible. Comparing figs. 2.7 and 2.8 we see that the CLAWS wordsketch is missing entire sections that are present in the TreeTagger version. Looking closer this is accounted for by CLAWS-tagged corpus’s frequency for convince-v\textsuperscript{1} is 2,182 — half of the 4,244 found by TreeTagger. Jumping over to the sketch for the adjective convince-j in the CLAWS corpus (fig. 2.9) seems to explain everything: TreeTagger conflates verbs and their adjectival forms, while CLAWS separates them. Another case where the aims of the underlying classification system must be kept in mind. In future work, the important thing will be to understand and control for appropriate features for each word class or synonym set under investigation.

\textsuperscript{1}(using SketchEngine/CQL’s \texttt{lempos} notation of the lemma followed by a one-character POS specifier)
Figure 2.7: Word Sketch of `convince-v`, in BNC tagged with CLAWS v5 tagger.
Figure 2.8: Word Sketch of *convinces*-v, in BNC tagged with TreeTagger 2.
CHAPTER 2. BACKGROUND

Figure 2.9: Word Sketch of `convinced-j`, in BNC tagged with CLAWS v5 tagger.
CHAPTER 2. BACKGROUND

Comparing word sketches side-by-side gets annoying fast, but it is also surprisingly useful. Useful enough that SketchEngine has a built-in comparison method to avoid the annoyance. Fig. 2.10 shows this for **persuade-v** and **convince-v**. SketchDiffs are tremendously informative and enriching, but retrieving them for every arbitrary pairing of words in a synset is laborious. Any given pairing of lemmas is of uncertain value or interest; other than similarity scores in its **thesaurus** function, SketchEngine has no way to guide its user toward the revealing pairings.
**Figure 2.10:** SketchDiff of **persuade** and **convinced**, in BNC tagged with CLAWS v5 tagger.
Chapter 3

Experimental Setup

The bones of this analytical framework are simply to collect a group of words to compare; to engineer features on which to compare them; cluster those feature sets; extract salient features by which to characterize the clusters as a whole; and extract salient features that are distinctive for each cluster member. Every step in that framework is so implementation-specific, and feature engineering so purpose-specific, the quality and usefulness of results is highly implementation-dependent. The key inflection points are feature selection, clustering algorithms, and the labeling of clusters.

3.0.1 Selection of words to compare

As nice as it would be to perform this analysis automatically over the whole lexicon, to prove the concept it will be more sensible to closely examine a set of synonyms under this framework and see how it goes. We chose a group of related meanings somewhat arbitrarily: verbs of convincing, persuading, or inducement.
CHAPTER 3. EXPERIMENTAL SETUP

Meanings under investigation

To limit the scope of experimentation to a tractably small set of features, we chose a set of verbs with related meanings around a core factor of inducement: persuade, convince, compel, and further-out associates like permit, allow, deny, refuse.

In FrameNet and VerbNet

FrameNet puts the core meanings in the Suasion frame: “A Speaker succeeds in getting an Addressee to believe or plan to execute the Content the Speaker has in mind.” The lexical units in FrameNet’s frame are convince.v, dissuade.v, incite.v, motivate.v, persuade.v, sway.v. FrameNet also has an Attempt_suasion frame, containing the verbs admonish.v, advise.v, advocate.v, badger.v, beg.v, cajole.v, enjoin.v, exhort.v, lobby.v, press.v, pressure.v, prevail.v, recommend.v, suggest.v, urge.v, wheedle.v.

FrameNet’s Frame Elements do not correspond in any particular way to the range of factors that may be of interest in disambiguating synonyms. At the Manner frame element, FrameNet’s frame description explains:

Any description of the event which is not covered by more specific FEs, including metaphorical force (hard, softly), secondary effects (quietly, loudly), and general descriptions comparing events (the same way). It may also indicate salient characteristics of a Manipulator that also affect the action (presumptuously, coldly, deliberately, eagerly, carefully).

‘Manner’ is a wide target, too wide to cover any specific distinguishers of meaning. Manners for ‘motivate’ are surely different from ‘dissuade’, or from ‘badger’ or ‘exhort’.

The VerbNet’s class that contains ‘convince’ is force-59. Its semantic roles are AGENT, PATIENT, and RESULT. It contains some nice synonyms like ‘coax’ and ‘prod’ but it also has things that seem farther afield from the central idea of ‘suasion’ that I sought to investigate: ‘commission’, ‘bullshit’, ‘palaver’. Since I was not doing my own feature engineering for
this iteration, I did not attempt to apply VerbNet’s mapping of syntactic frame patterns to semantic argument formulas, but doing so could have quite interesting results in the future.

**My characterization of the meanings under investigation**

After collecting a list of words, but before performing any of the clustering or labeling, I worked through my own intuitive set of semantic factors present in these verbs of persuasion. What follows is a description of these elements, which I hoped to find through my automatic analysis.

An agent with some degree of desire of an outcome, and a patient with their own degree of desire of said outcome. In *encourage*, both agent and patient want the outcome; in *force*, the agent’s desire is strong, while the patient’s desire is negative; in *permit*, the agent is somewhere between neutral on, or somewhat opposed to, the outcome, while the patient is inclined to want it.

The object of the desired outcome may be an action (including *inaction* in e.g. *warn, forbid*) or an attitude towards a proposition. The traditional distinction between *persuade* and *convince* concerns two factors: first, the nature of the outcome (an action for *persuade*, a proposition for *convince*); second, the firmness of the resulting opinion-state, which firmness is emphasized in *convince*.

The manner or means by which the outcome is sought may vary: *convince* and *persuade* require speech activity on the part of the agent; *encourage* is also usually accomplished by communication, but of a less argumentative sort than *persuade*. *Permit* and *allow* essentially require inaction or forbearance from stopping someone’s doing something; some verbs like *manipulate* may be to induce someone to do something without their complete awareness of the situation.

Another factor is relative strength of the urging: urgeativity. It is normal to ‘strongly
urge’ someone, but not ‘strongly enable’ or ‘strongly reassure’. Propose and suggest are rather noncommittal and measured. Strength of urging is also related to agent’s concern for willingness in the patient: a proposer is hoping for the patient to reach agreement almost through their own line of reasoning. Those who force and compel would see the outcome even with the patient’s desires unchanged; persuade and convince, if successful, have changed the patient’s internal state to some degree.

Temporal dimensions are relevant to the verbs of seeing in [Pustejovsky and Joshi, 2017] — e.g. glance vs. stare — and while they do not bring to mind any obvious relevance to verbs of inducement, they may nevertheless be a factor. ‘We finally convinced them’ indicates that conviction may be a gradual process; ‘in the end we were forced to comply’ suggests force as a last resort. By contrast, tempt or encourage seem to resist temporal modification: ‘quickly encouraged’ doesn’t have any particular intuitive salience. ‘Quickly reassure’ suggests a picture where reassurance is quickly offered — and calls our attention to the fact that reassurance, like encouragement, does not entail that an agent’s intended change of opinion-state actually takes place in the patient. That is, you can reassure someone all you want, and they might still be anxious.

3.0.2 Feature extraction: Sketch Engine Gramrels

As a shortcut to feature engineering, I fetched word sketches in JSON format from the Sketch Engine’s BNC corpora. There are multiple instances of BNC in Sketch Engine, with different taggers and different Sketch Grammars. Of particular interest is the preloaded/bnc2 corpus, which as of the time of our data fetch was tagged with the CLAWS tagger. The grammar extracts a block of ‘usage patterns’ which extracts very Apresjanian unary factors of lemma occurrences: like appearance in formulations like reflexive (‘cannot convince myself’), passive (‘she was utterly convinced’), np.VPto (‘convince me to change my mind’) or
For this thesis I compare three approaches to extracting features from Word Sketches:

- only the adverbial modifiers of verbs, both as words themselves and generalized to Roget hyperonyms, in CLAWS BNC instances;
- only ‘usage patterns’ in the CLAWS-tagged corpus;
- the full set of gramrels provided by the CLAWS word sketch

Reducing dimensionality of adverbial modifiers

For every modifying adverb, I extracted both the word itself and its Roget hyperonym. This was intended to reveal both generalities (say, a free selection among modifiers concerned with time or emotion) and specificities (when there is a somewhat fixed single option for modification). There are too many adverbs to treat them all as completely separate features, and doing so would ignore the underlying semantic structure that I’m trying to make generalizations about.

[Jarmasz and Szpakowicz, 2001] give a thorough comparative evaluation of Roget and WordNet, as a means of introducing their excellent digitization of Roget into an electronic knowledge base. [Kennedy and Szpakowicz, 2014] provide an automatically-updated version of the 1911 Roget at Open Roget’s, and this is what I have used as a knowledge base for the present work. Since the rest of my pipeline is in Python, I have written a Python module to parse both the hierarchical information and synonym-membership lists necessary to classify collocates by their semantic class.

For purposes of dimensionality reduction, Roget offers another distinct advantage: a mere 1,000 classes, in contrast with WordNet’s 117,000+ synsets. This is by no means an apples-to-apples comparison: it is standard to exploit Roget’s paragraph and semicolon
groups for finer-grained associations within a Roget heading, but even down to the level of
semicolon groups, Roget’s 60,000 semicolon groups come to half of WordNet’s synset count,
while Roget’s 220,000+ strings exceed WordNet 3.0’s 206,000+.

At any rate, the purpose of choosing Roget over WordNet is to drive for dimensionality
reduction in the feature space, and semantic distinctness and granularity in the meaning
space – not headword count. The hierarchical structure of Roget is only usable down to the
heading level: at semicolon level and lower, word-word relations can reliably be interpreted
only along a long gradient from ‘general’ to ‘specific’ instances of the heading’s class.

In the many cases where a word was entered at several points in the Roget hierarchy,
I exploited the structure of the classification to find the most ‘core’ synonym in the fol-
lowing way. OpenRoget assigns an ordinal integer for each word’s position in a paragraph,
semicolon-group, and comma-group. My intuition was that the earlier a word appeared in a
Roget entry, the closer it was in meaning to the heading of the paragraph. This was an easy
path to consistently choosing a single Roget path for a given word. As we will see, this had
some quirks: most importantly, I did not control for part of speech, so the adverb still was
taken to be a noun hyponymous to vaporizer. Future iterations will fix this.

3.0.3 Clustering techniques: Affinity Propagation, K-Means

I used the Scikit-Learn implementations of two clustering algorithms. Affinity propagation
[Frey and Dueck, 2007] creates hierarchical clusters with an induced number of clusters;
K-means requires selection of a value of K to determine its number of clusters. In order to
have somewhat comparable clusters, I clustered first with AP and then used the number of
clusters found by AP as the K for K-Means. For K-Means, I used a constant random seed
so that repeated runs of the algorithm would reflect differences in feature selection, rather
than differences in the initialization of randomness.
I experimented with various value matrices as the basis of clustering. Underlying everything is a matrix of verbs of interest and features associated with them, normalized with TF-IDF (where a ‘document’ is a word, and a ‘term’ is a feature). Clustering on a correlation, i.e. the dot-product of the TF-IDF matrix and its transpose, yielded nearly identical clusters in both AP and K-Means. The input matrices were pretty low-dimensionality: 21 \times 21 in my 21-word handpicked sample. Clustering on TF-IDF (a matrix with a shape of 21 \times 700–1000 depending on the features extracted) yielded clusters that were largely similar to the correlation clusters and also to one another, which was reassuring: but these offered enough differences that they were interesting to compare.

### 3.0.4 Salient feature extraction: CF-IDF and beyond

To ‘label’ the clusters with the distinguishing factors they tended to be associated with, I worked through several iterations of extracting salient features. Initially I used the principles of TF-IDF to extract the most salient features for each cluster \(C\) and their members \(c_1...c_n\). For characterizing clusters, the ‘TF’ term is really ‘Cluster Frequency’, i.e. the overall frequency of each feature \(r\) within the members of the cluster \(c\); IDF is the inverse document frequency as usual:

\[
w_{r,C} = \log(cf_{r,C}) \times \log\left(\frac{N}{df_r}\right)
\]

The top \(N\) ‘distinguishing factors’ for a cluster are then the items with the top \(N\) CF-IDF scores. It quickly became clear, however, that this formula over-privileged high-frequency collocations that were only strongly associated with a single, usually highest-frequency term in a cluster.

To dampen this effect, I did a simple sort of normalization: divide the total \(cf_{r,C}\) by (the number of items in \(c_1...c_n\) that lack feature \(r\)) + 1. Adding 1, so that if a feature is present

30
for all cluster members, it is credited with its full count; if a feature is missing from only one
member, its count for the cluster is halved, and so on. This can be expressed as

\[ dampen(r, C) = |\{c | c \in C \land tf(r, c) = 0\}| \]

So for each cluster C:

\[ w_{r,C} = \log\left(\frac{cf_{r,C}}{|C| + 1 - dampen(r, C)}\right) \times \log\left(\frac{N}{df_r}\right) \]

Tables 4.1 and 4.2 show clusters generated on a TF-IDF matrix where the features are
adverbial modifiers (indicated with ‘w=’) and their Roget hyperonyms (↑). Table 4.1 is
labeled with the raw TF-IDF labels and table 4.2 with the normalized TF-IDF.

In the results tables below, each cluster is labeled by its top 10 features by this formula.
There is an unmistakable effect of the highest-frequency collocations, and in future work
I must make a concerted effort to normalize. My labeling is rudimentarily inspired by
[treeratpituk and callan, 2006] but does not attempt to be an complete implementation
of their techniques: in future work I will draw more carefully from these and other cluster-
labeling methods.
Chapter 4

Results

The semantic factors that we have found through these first experiments appear highly subjective, and on a different level than most of those found by factorization techniques previously. They do point to intensional factors that are commonly made overt in prose. They require a fair amount of interpretation, so they are not immediately usable for computational purposes. But, happily, they yield to interpretation rather well, so there may be a reasonably short path to using them for lexical analysis and lexicography of a sort that could make resources usable for computational purposes.

4.1 Adverbial Modifiers Only

4.1.1 Interpretation of clusters and labels

Looking at the ‘Synonyms’ column of table 4.1, interpreting the differences in groupings intuitively, we see some noteworthy things.

Demand/refuse: K-means cluster 7 contains ‘demand, refuse’, but in Affinity Propagation these are singletons in clusters 1 and 6. When they are paired, the un-normalized labels
CHAPTER 4. RESULTS

<table>
<thead>
<tr>
<th>No.</th>
<th>Synonyms</th>
<th>Top adverbial modifiers from CLAWS, simple TF-IDF labeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>compel, convince, enable, permit</td>
<td>↑in the circumstances, w=thus, ↑eventually, w=eventually, ↑through, ↑both, w=thereby, ↑in return, ↑wholly</td>
</tr>
<tr>
<td>2</td>
<td>advocate, propose, suggest</td>
<td>w=here, ↑essaying, w=al, ↑on high, ↑new, w=tentatively, ↑seriously, ↑prototype, w=originally, w=previously</td>
</tr>
<tr>
<td>3</td>
<td>claim, request</td>
<td>↑intellect, w=reasonably, w=falsely, ↑conditional, w=afterwards, ↑formally, ↑rightly, ↑subsequently, w=justifiably, w=rightly</td>
</tr>
<tr>
<td>4</td>
<td>advise, encourage, instruct</td>
<td>w=ill, w=best, ↑photograph, ↑humbly, ↑evil, ↑actively, ↑stock, w=well, w=positively, ↑best(677)</td>
</tr>
<tr>
<td>5</td>
<td>force, let, persuade</td>
<td>w=right, ↑complement, w=up, w=finally, ↑separate, ↑finally, w=out, w=through, w=down, ↑down</td>
</tr>
<tr>
<td>6</td>
<td>oblige, remind, tell, urge</td>
<td>↑on time, w=exactly, ↑render insensible, w=about, w=bluntly, w=there, w=much, w=today, ↑around, w=constantly</td>
</tr>
<tr>
<td>7</td>
<td>demand, refuse</td>
<td>↑resolutely, w=point-blank, w=resolutely, w=steadfastly, ↑angrily, w=stubbornly, w=angrily, w=obstinately</td>
</tr>
</tbody>
</table>

AP

<table>
<thead>
<tr>
<th>No.</th>
<th>Synonyms</th>
<th>Top adverbial modifiers from CLAWS, simple TF-IDF labeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>demand</td>
<td>w=silently, w=sharply, w=irritably, ↑material, ↑angrily, w=angrily, w=increasingly, ↑vast, w=physically, ↑accusing</td>
</tr>
<tr>
<td>2</td>
<td>compel, convince, enable, encourage, permit, persuade</td>
<td>↑photograph, ↑actively, ↑easily, ↑wholly, w=actively, w=easily, ↑in the circumstances, w=positively, w=thus</td>
</tr>
<tr>
<td>3</td>
<td>advise, instruct, request</td>
<td>w=ill, ↑humbly, ↑conditional, ↑best(677), ↑evil, w=best, ↑special, ↑stock, w=well, w=specifically</td>
</tr>
<tr>
<td>4</td>
<td>force, let, tell</td>
<td>w=ever, w=out, ↑separate, ↑obtrude, w=through, w=off, w=in, ↑down, ↑far off, w=down</td>
</tr>
<tr>
<td>5</td>
<td>oblige, propose, suggest</td>
<td>↑without, w=seriously, w=tentatively, ↑prototype, ↑essaying, w=recently, ↑new, w=above, w=originally, w=al</td>
</tr>
<tr>
<td>6</td>
<td>refuse</td>
<td>↑persevering, w=flatly, ↑resolutely, w=stubbornly, w=steadfastly, w=absolutely, w=first, w=point-blank, w=resolutely, w=blank</td>
</tr>
<tr>
<td>7</td>
<td>advocate, claim, remind, urge</td>
<td>↑publicly, w=afterwards, ↑rightly, ↑subsequently, w=publicly, w=rightly, ↑reason ill, w=today, ↑legitimate, w=falsely</td>
</tr>
</tbody>
</table>

Table 4.1: Clusters and top CF-IDF labels from the handpicked subset of close synonyms with CLAWS v5

like resolutely, point-blank, angrily, stubbornly seem to have intuitive salience for both of these words: we can detect a low concern for the desires of the patient of the refusal/demand, and the agent’s strong attachment to the desired outcome. ‘Point-blank’ reflects that there is not necessarily much propositional content to the refusal: it is almost a speech act.

When these two verbs are separated in the AP clustering, the un-normalized labels some nuance in this story: refusal can be persevering, flat, stubborn, but it is demand that is most
### Chapter 4. Results

<table>
<thead>
<tr>
<th>No.</th>
<th>Synonyms</th>
<th>Top adverbial modifiers from CLAWS, cluster-normalized CF-IDF labeling</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>w=thereby, ↑vaporizer, w=thus, w=still, ↑exact, ↑never, w=so, w=never, ↑in the circumstances</td>
</tr>
<tr>
<td>1</td>
<td>compel, convince, enable, permit</td>
<td>w=never, ↑in the circumstances</td>
</tr>
<tr>
<td>2</td>
<td>advocate, propose, suggest</td>
<td>w=recently, w=previously, w=seriously, w=here, w=originally, ↑instead, ↑here, ↑prototype, w=instead</td>
</tr>
<tr>
<td>3</td>
<td>claim, request</td>
<td>w=rightly, w=subsequently, ↑before, w=further, ↑intellect, ↑relieve, w=reasonably</td>
</tr>
<tr>
<td>4</td>
<td>advise, encourage, instruct</td>
<td>↑strongly, ↑please, w=strongly, w=please, ↑relieve, ↑stock, w=well, w=further</td>
</tr>
<tr>
<td>5</td>
<td>force, let, persuade</td>
<td>↑complement, w=through, w=out, ↑obtrude, ↑separate, ↑down, w=in, w=down, w=finally, ↑finally</td>
</tr>
<tr>
<td>6</td>
<td>oblige, remind, tell, urge</td>
<td>w=again, w=always, w=today, ↑right, ↑the present, w=just, ↑habitually(16), w=constantly, ↑repeatedly, w=repeatedly</td>
</tr>
<tr>
<td>7</td>
<td>demand, refuse</td>
<td>w=blank, w=repeatedly, w=yesterday, ↑beginning, ↑yesterday, w=initially, w=angrily, ↑angrily</td>
</tr>
</tbody>
</table>

### Table 4.2: Clusters and top normalized CF-IDF labels from the handpicked subset of close synonyms with CLAWS v5

likely to be *sharp, angry, irritable.*

The normalized labeling takes us a bit farther. Singleton clusters have the same labels in both methods, but the labels on the pair at K-means #7 in table 4.2 shows the most salient adverbs used with both *demand* and *refuse*: *[point]-blank, repeatedly, yesterday, initially, angrily.* *Initially* is interesting, in contrast with the steadfastness we see elsewhere: it is possible for demands and refusals to be taken as merely a starting point.
CHAPTER 4. RESULTS

Oblige moves between K-means cluster 6 (with *remind, tell, urge*) and AP cluster 5 (with *propose, suggest*). The similarity among most of the members of these clusters is pretty clear – ‘tell’ being a pretty unmarked vehicle for either propositional content or a request for action, and *urge, propose, suggest, remind* gaining a few additional elements — and it really *oblige* that doesn’t fit well with either of them. For many of my experiments, *oblige* actually ended up in a singleton cluster, and in retrospect it probably does not belong in a Suaision class anyway. On its own, *oblige* occurs mostly in senses like ‘much obliged’ or ‘kindly obliged’ which is definitely far removed from suasion. To only deal with senses like ‘legally obliged to do X’, we will need a more nuanced approach to polysemy.

Convince/persuade There is a classic prescriptive distinction between *convince* and *persuade*, where *convince* has a result of ‘complete conviction’ but *persuade* could mean ‘reluctant agreement’. This distinction can perhaps be detected in the classification of *convince* in K-means #1 and AP #2, contrasted with the placement of *persuade* in K-means #5 and AP #2. K-means cluster #1 has ‘wholly’ as one of its salient labels in the un-normalized labeling of table 4.1, but this is missing from the normalized table 4.2 — meaning that it is particularly marked for only a few members of the cluster, so it is likely the contribution of *convince*.

Even more intriguing, if you squint at it right, are the other remaining modifiers in K-means #1: *thereby, thus, eventually*. To *convince*, as well as to *enable, compel*, may be something that requires a means of convincing/compulsion: we are convinced by arguments, compelled by law or circumstance. Contrast with *demand/refuse*: there is no habitual means of demanding. A demand can be expressed as a proposition, but conviction and compulsion are a resulting state of something.

K-means cluster 3 *advise, claim, encourage, instruct, request* is a superset of AP cluster 3 *advise, instruct, request* plus two words that are in different AP clusters. The
presence of *falsely* in K-means #3 seems to be the contribution of *falsely claim*; the presence of *ill* (and also ↑*evil* in AP #3) seems to be the contribution of *ill advised*.

K-means cluster #2 and AP #5 contain **propose, suggest**. The salient presence of *seriously* calls to mind another feature we could be looking for: ‘are you seriously proposing...?’ or ‘you’re not seriously suggesting that...?’ convey the questioner’s suggestion that the proposer/suggester is in fact unserious. We should at least extract a feature for whether a sentence ends in a question mark or if the verb is in a dependency relation with a negation. But we also see *tentatively* and ↑*essaying* — reflecting the low urgeativity of these two verbs, in contrast with ↑*tentatively compel* or ↑*tentatively demand*. ↑without is an interesting feature that I would like to examine further — it may match some very tentative formulations like ‘without suggesting wrongdoing...’ or ‘without proposing a theoretical analysis...’, or it may be another quirk of the Roget featurization.

Also worth considering with **propose, suggest** are *originally, recently* and the superordinate ↑*prototype*. Contrast with *eventually convince* or *finally force*: proposal and suggestion can be the beginning of a process of suasion, while conviction is one possible outcome of that process (and the use of force another: Louis XIV’s *ultima ratio regum*).

### 4.2 Usage Patterns

SketchEngine’s CLAWS-tagged BNC instance has defined 26 unary properties of verb occurrences that can be extracted as ‘usage patterns’. These seem to be a good initial attempt at discovering what Apresjan [Apresjan, 2000, p. 46–50] terms ‘syntactic distinctions’: grammatical features of a word’s occurrence that are syntactic or morphological in nature. It is comparatively rare for these kinds of features to show up in corpus-lexicographic tools, but they have the potential to inform very refined stylistic choices. It was a pleasant surprise to see them in the CLAWS BNC in SketchEngine. Unfortunately they are a little unscrutable,
CHAPTER 4. RESULTS

Figure 4.1: Sketch Grammar definitions of some ‘usage pattern’ features requiring interpretation of a very different sort than the adverbial modifiers in the preceding section. A sample of the definitions of these patterns is shown in Figure 4.1.

With only 26 usage-pattern features defined in the grammar, it turned out that my set of 21 close synonyms exhibited 24 of them, and nearly all of the verbs seem to match all of the common grammatical patterns. As a result, the ‘raw CF-IDF’ and ‘normalized’ labeling functions had identical results for the top 10 usage patterns in each cluster. So table 4.3 shows only the raw CF-IDF labels.

Two features that particularly interest me are reflexive and passive. My intuition tells me to expect convince oneself to be salient, but in the groupings and labelings of table 4.3, reflexive is not salient for the cluster that contains convince by either clustering algorithm. The only cluster for which reflexive comes out as highly salient is the singleton cluster containing let: ‘you let yourself get walked over’. The reflexive feature is in the top 5 for convince on its own, but it is not part of the main pattern for convince’s cluster by these algorithms.
CHAPTER 4. RESULTS

Table 4.3: Clusters and top raw CF-IDF labels from the handpicked subset of close synonyms with CLAWS

<table>
<thead>
<tr>
<th>No.</th>
<th>Synonyms</th>
<th>Top ‘usage patterns’ from CLAWS, raw CF-IDF labeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>enable, oblige, persuade, urge</td>
<td>prep_wh, prep_Sing, np_pp, np_sfin, np_np, passive, quote, prep_ing, np_VPing, np_adv</td>
</tr>
<tr>
<td>2</td>
<td>convince, tell</td>
<td>np_pp, np_np, np_adv, prep_ing, quote, np_sfin, passive, np_VPing, np_VPto, it+</td>
</tr>
<tr>
<td>3</td>
<td>advise, compel, encourage, force, instruct, permit, propose, refuse, request</td>
<td>np_np, np_adv, passive, np_sfin, prep_ing, quote, np_VPto, np_VPing, np_VPbare, it+</td>
</tr>
<tr>
<td>4</td>
<td>let</td>
<td>passive, np_pp, np_np, prep_ing, np_sfin, np_VPing, quote, np_adv, reflexive, np_VPto</td>
</tr>
<tr>
<td>5</td>
<td>advocate, demand, remind</td>
<td>np_sfin, np_pp, np_np, passive, prep_ing, quote, np_adv, np_VPing, np_VPto, np_VPbare</td>
</tr>
<tr>
<td>6</td>
<td>claim, suggest</td>
<td>passive, np_np, np_VPto, np_sfin, prep_ing, quote, np_adv, np_VPing, Scond, np_VPbare</td>
</tr>
<tr>
<td>AP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>claim, suggest</td>
<td>passive, np_np, np_VPto, np_sfin, prep_ing, quote, np_adv, np_VPing, Scond, np_VPbare</td>
</tr>
<tr>
<td>2</td>
<td>compel, force, refuse</td>
<td>np_VPbare, prep_wh, np_pp, np_np, np_sfin, passive, quote, prep_ing, np_VPing, np_adv</td>
</tr>
<tr>
<td>3</td>
<td>let</td>
<td>passive, np_pp, np_np, prep_ing, np_sfin, np_VPing, quote, np_adv, reflexive, np_VPto</td>
</tr>
<tr>
<td>4</td>
<td>advise, encourage, instruct, permit, propose, request</td>
<td>np_pp, np_np, passive, np_sfin, prep_ing, quote, np_adv, np_VPing, np_VPto, np_VPbare</td>
</tr>
<tr>
<td>5</td>
<td>enable, oblige, persuade, urge</td>
<td>prep_wh, prep_Sing, np_pp, np_sfin, np_np, passive, quote, prep_ing, np_VPing, np_adv</td>
</tr>
<tr>
<td>6</td>
<td>advocate, convince, demand, remind, tell</td>
<td>np_adv, passive, np_sfin, prep_ing, quote, np_VPto, np_VPing, np_VPbare, it+, VPto</td>
</tr>
</tbody>
</table>

Affinity Propagation’s cluster #6 is the union of K-means’s clusters #2 and #5. Intuitively, almost all of these verbs — advocate, convince, remind, tell — tend to carry propositional content (‘convince someone of a proposition’, ‘tell someone a proposition’); demand is the odd one out, perhaps as likely to be a demand for action as for any sort of information. What usage features distinguished them as a cluster? Convince and tell are strongly associated with both the presence of a prepositional adjunct (np_pp) and a noun phrase after the verb (np_np). Both of these are exhibited in a standard ditransitive phrasing like ‘they_np convinced her_np of_pp his avowed antipathy’.  

38
Although these same np_np and np_pp features are also highly salient for advocate or demand, they do occur in different proportion and also, intriguingly, in different kinds of argument structure vis-à-vis the verbs. The labeling also surfaces entirely different top features for AP cluster #6, the union of the two K-means features: there it is adverbial modification (np_adv) and passive that are most salient for the whole cluster. What to make of this? Hard to say. Looking at the word sketches for the individual words in the cluster, it turns out that np_adv is listed as salient only for tell — so this may actually be an artifact of my normalization method. The feature seems to be targeted at noun phrase adverbials — ‘told the lieutenant over at the hotel’, ‘I’ll tell you over dinner’, ‘told his tale out of spite’.

The usage patterns in the word sketches are very suggestive of the potential of features that are not pure dependencies, but to be really useful they surely need to be engineered from scratch with specific questions in mind. A fruitful possibility for future work, with Apresjan and other theorists in hand.

### 4.3 Full-wordsketch features

The bulk of the work for this thesis was focused on adverbial modification. But the same framework that ingests adverb features can just as well ingest any other features. So I also threw the full set of wordsketch features through this system, to see what would come out. Looking up the semantic superordinates of adverbs required a bit of hand-tuning, beyond what was feasible to quickly do for the full set of grammatical relations, so the following tables are not aimed at semantic generalities. Table 4.4 shows the clusters with simple TF-IDF labeling, table 4.5 with normalized labels.

A few things quickly become obvious. Subjects and objects are the overwhelmingly salient gramrels for most of these verbs: ‘injunction compelling the union to call off the picket’, ‘European countries will be compelled to adopt WHO guidelines’. Occasionally other
### CHAPTER 4. RESULTS

<table>
<thead>
<tr>
<th>No.</th>
<th>Synonyms</th>
<th>Top wordsketch gramrels from CLAWS, simple TF-IDF labeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>compel, oblige</td>
<td>object=union-n, object=feeling-n, modifier=logically-a, object=country-n, object=witness-n, pp_by-p=law-n, object=person-n, object=council-n</td>
</tr>
<tr>
<td>2</td>
<td>convince, enable, encourage, persuade, remind</td>
<td>object=gentleman-n, object=teacher-n, object=use-n, modifier=positively-a, object=pupil-n, subject=page-n, subject=system-n, object=child-n</td>
</tr>
<tr>
<td>3</td>
<td>claim, demand, instruct, refuse, request, suggest</td>
<td>subject=analysis-n, object=need-n, object=victory-n, object=solicitor-n, object=credit-n, object=right-n, object=request-n, subject=theory-n</td>
</tr>
<tr>
<td>4</td>
<td>force, let, tell</td>
<td>object=smile-n, object=dad-n, object=mp-n, object=eye-n, object=inquest-n, object=magistrate-n, object=newspaper-n, object=joke-n</td>
</tr>
<tr>
<td>5</td>
<td>advocate, permit, propose</td>
<td>subject=weather-n, subject=bill-n, pp_by-p=commission-n, object=measure-n, object=analysis-n, subject=commission-n, subject=party-n, subject=question-n</td>
</tr>
<tr>
<td>6</td>
<td>advise, urge</td>
<td>subject=night-n, object=caution-n, part_trans=on-a, object=client-n, subject=manager-n, pp_on-p=policy-n, object=horse-n, modifier=ill-a</td>
</tr>
<tr>
<td>AP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>compel, demand, oblige, permit</td>
<td>object=development-n, and/or=supply-v, object=resignation-n, object=cash-n, object=fee-n, object=respect-n, object=payment-n, subject=situation-n</td>
</tr>
<tr>
<td>2</td>
<td>force, let, suggest</td>
<td>object=reason-n, object=dog-n, object=hair-n, part_intrans=down-a, object=possibility-n, subject=analysis-n, object=need-n, part_intrans=in-a</td>
</tr>
<tr>
<td>3</td>
<td>advise, convince, enable, encourage, persuade, remind, urge</td>
<td>object=teacher-n, object=use-n, object=researcher-n, modifier=positively-a, subject=page-n, object=child-n, subject=option-n, object=individual-n</td>
</tr>
<tr>
<td>4</td>
<td>advocate, claim, instruct, propose</td>
<td>subject=paper-n, object=resolution-n, object=theory-n, object=motion-n, object=toast-n, object=change-n, object=model-n, object=responsibility-n</td>
</tr>
<tr>
<td>5</td>
<td>refuse, request</td>
<td>object=help-n, object=assistance-n, object=admission-n, object=entry-n, object=bail-n, and/or=grant-v, pp_on-p=grounds-n, modifier=flatly-a</td>
</tr>
<tr>
<td>6</td>
<td>tell</td>
<td>object=Commons-n, object=friend-n, object=scientist-n, subject=instinct-n, object=mother-n, object=reporter-n, object=newspaper-n, object=inquest-n</td>
</tr>
</tbody>
</table>

Table 4.4: Clusters and top simple CF-IDF labels from handpicked synonyms with all wordsketch gramrels as features

gramrels peek through: ‘obliged by law to pay into a pension fund’. Reading the selection of gramrels in tables 4.4 and 4.5, the overall impression gets to be rather newspaperish. There are exceptions: ‘forced a smile’, ‘a forced joke’, which may be literary instead. The puzzling
### CHAPTER 4. RESULTS

<table>
<thead>
<tr>
<th>No.</th>
<th>Synonyms</th>
<th>Top wordsketch gramrels from CLAWS, normalized TF-IDF labeling</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>K-means</strong></td>
</tr>
<tr>
<td>1</td>
<td>compel, obliged</td>
<td>object = union-n, object = witness-n, object = attention-n, object = person-n, object = feeling-n, object = country-n, object = council-n, object = company-n</td>
</tr>
<tr>
<td>2</td>
<td>convince, enable, encourage, persuade, remind</td>
<td>modifier = thus-a, object = public-n, object = member-n, object = individual-n, subject = policy-n, object = teacher-n, object = development-n, object = pupil-n</td>
</tr>
<tr>
<td>3</td>
<td>claim, demand, instruct, refuse, request, suggest</td>
<td>object = credit-n, object = victory-n, object = support-n, subject = theory-n, object = application-n, object = money-n, object = information-n, object = way-n</td>
</tr>
<tr>
<td>4</td>
<td>force, let, tell</td>
<td>object = inquest-n, part_intrans = down-a, object = truth-n, object = tale-n, part_trans = down-a, object = story-n, object = reporter-n, object = police-n</td>
</tr>
<tr>
<td>5</td>
<td>advocate, permit, propose</td>
<td>object = solution-n, modifier = originally-a, object = reform-n, subject = question-n, subject = weather-n, object = establishment-n, object = legislation-n, subject = commission-n</td>
</tr>
<tr>
<td>6</td>
<td>advise, urge</td>
<td>object = chancellor-n, object = government-n, object = motorist-n, and/or = help-v, object = plaintiff-n, object = client-n, modifier = ill-a, pp_on-p = matter-n</td>
</tr>
</tbody>
</table>

|     |          | **AP**                                                        |
| 1   | compel, demand, obliged, permit | pp_by-p = law-n, object = respect-n, subject = situation-n, pp_for-p = money-n, object = return-n, subject = act-n, object = analysis-n, modifier = physically-a |
| 2   | force, let, suggest | subject = analysis-n, object = need-n, object = dog-n, part_trans = down-a, part_trans = in-a, subject = theory-n, object = possibility-n, subject = study-n |
| 3   | advise, convince, enable, encourage, persuade, remind, urge | object = manager-n, object = secretary-n, object = individual-n, object = teacher-n, subject = system-n, object = client-n, object = people-n, subject = option-n |
| 4   | advocate, claim, instruct, propose | object = measure-n, object = toast-n, subject = party-n, object = motion-n, object = policy-n, subject = company-n, object = model-n, object = support-n |
| 5   | refuse, request | object = admission-n, object = assistance-n, and/or = grant-v, object = licence-n, object = leave-n, pp_on-p = grounds-n, object = bail-n, modifier = flatly-a |
| 6   | tell | object = Commons-n, object = friend-n, object = scientist-n, subject = instinct-n, object = mother-n, object = reporter-n, object = newspaper-n, object = inquest-n |

Table 4.5: Clusters and top normalized CF-IDF labels from handpicked synonyms with all wordsketch gramrels as features

Item in Table 4.4’s K-means cluster #6, where ‘night’ is a salient subject of ‘urge’, turns out to be the phrase ‘so-and-so last night urged’ — really a failing of the sketch grammar to recognize an adverbial adjunct phrase.
4.3.1 Subjects and objects smell like their genres

Reading through these clusters in the aggregate, each pairing of subject-verb or object-verb carries with it an unmistakable sense of the genre from which it must be taken, even without consulting the corpus: ‘compelling a witness’; ‘urged the chancellor’; ‘tell the inquest’ vs. ‘tell your mum’; ‘propose a measure’ vs. ‘propose a toast’. The notion that high-salience subject and object relations carry a strong odor of the text types from which they are taken, while adverbial modifiers feel far less marked for being from a particular genre, is the most fascinating and surprising thing I’ve found in my work on this thesis. I will elaborate and speculate on this in my concluding chapter.

On one hand, this points to a need for an entirely different approach than Roget- or WordNet-derived semantic generalization. We could collapse union to Organization, witness to (unfortunately) Spectator, but that is not a path to the kind of generalization that is useful for the kinds of semantic questions we want to ask. To make generalizations that are linguistic and lexical, rather than just profiling of genres, we may need non-ontological resources. It is possible that the semantic types for nominals in CoreLex [Buitelaar, 1998] would provide the kinds of features that promote these generalizations, and that is an exciting direction for future development of this work.

4.3.2 Filtering out subjects and objects

To see what there is to be seen with the nominal relations out of the way, table 4.6 shows the same data as 4.5 but with the subject and object relations filtered out.

These end up being rather sparse: K-means cluster #3 has only one element, the adverbial steadfastly that we have seen in the adverbial section. Most of the interesting things here are prepositional-phrase arguments (‘compelled on pain of such-and-such’, ‘obliged by law’, ‘proposed by the commission’), and these are usually adjuncts, although K-means #4 also
### CHAPTER 4. RESULTS

<table>
<thead>
<tr>
<th>No.</th>
<th>Synonyms</th>
<th>Top non-subject/object gramrels from CLAWS, normalized TF-IDF labeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means</td>
<td>compel, oblige</td>
<td>modifier=however-a, modifier=greatly-a, modifier=nevertheless-a, pp_on-p=pain-n, pp_by-p=law-n, modifier=logically-a, modifier=legally-a, modifier=much-a</td>
</tr>
<tr>
<td>1</td>
<td>convinc, enable, encourage, persuade, remind</td>
<td>modifier=greatly-a, and/or=advertise-v, and/or=support-v, modifier=easily-a, pp_by-p=success-n, modifier=positively-a, modifier=actively-a</td>
</tr>
<tr>
<td>3</td>
<td>claim, demand, instruct, refuse, request, suggest</td>
<td>modifier=steadfastly-a</td>
</tr>
<tr>
<td>4</td>
<td>force, let, tell</td>
<td>part_out-a_obj=breath-n, part_intrans=out-a, part_intrans=in-a, part_intrans=down-a, modifier=there-a, part_trans=down-a, part_trans=off-a, part_trans=out-a</td>
</tr>
<tr>
<td>5</td>
<td>advocate, permit, propose</td>
<td>modifier=long-a, pp_by-p=party-n, pp_in-p=bill-n, and/or=require-v, modifier=originally-a, pp_by-p=commission-n, and/or=cause-v</td>
</tr>
<tr>
<td>6</td>
<td>advise, urge</td>
<td>modifier=best-a, modifier=properly-a, pp_on-p=aspect-n, pp_on-p=matter-n, and/or=help-v, modifier=well-a, modifier=strongly-a, pp_on-p=matter-n</td>
</tr>
</tbody>
</table>

| AP | compel, demand, oblige, permit | pp_for-p=product-n, modifier=duly-a, modifier=angrily-a, modifier=physically-a, modifier=usual-a, and/or=require-v, pp_for-p=money-n, and/or=supply-v |
| 1 | force, let, suggest | modifier=above-a, part_intrans=in-a, part_intrans=down-a, part_intrans=out-a, part_out-a_obj=breath-n, part_trans=down-a, part_trans=in-a, part_trans=off-a |
| 3 | advise, convince, enable, encourage, persuade, remind, urge | and/or=help-v, and/or=support-v, modifier=best-a, pp_by-p=success-n, modifier=actively-a, modifier=positively-a, modifier=well-a |
| 4 | advocate, claim, instruct, propose | modifier=than-a, modifier=originally-a, pp_by-p=commission-n |
| 5 | refuse, request | modifier=consistently-a, modifier=unreasonably-a, modifier=blank-a, modifier=specifically-a, pp_to-p=back-n, modifier=resolutely-a, modifier=absolutely-a, modifier=first-a |
| 6 | tell | and/or=go-v, modifier=there-a, part_trans=off-a, modifier=exactly-a, part_intrans=off-a, modifier=ever-a, and/or=come-v |

Table 4.6: Clusters and top normalized CF-IDF labels from handpicked synonyms with non-subject/object gramrels as features

shows some prepositional-phrasal verbs whose PP object, if present, is sometimes the direct object: ‘forced out a breath’, ‘let off’, ‘tell off’. These may be sometimes salient, sometimes irrelevant to our core semantic question. Different approaches to generalization for each grammatical relation (or at least for each part of speech) will be revealing.
Chapter 5

Conclusion

5.1 Observations

This is an encouraging start.

If my focus had been purely lexicographical, I might have positioned this thesis as a new lexicographical tool, perhaps the ‘SynSetSketch’. I am certain that this approach, with refined and focused feature extraction, can be of use to lexicographers in creating new lexical resources more efficiently. But I also believe that there is broader theoretical progress, not just descriptive efficiency, to be made through lexical analysis in this framework.

Drawing on my past experience in lexicography, I see strong indications that this approach has, at the bare minimum, the potential to accelerate lexicographical work on synonym groups. After looking carefully at a few dozen tables of results, I have grown to understand where and how to look, how to blur my semantic vision to see generalities that hadn’t been completely featurized, and to reach an understanding of a synonym set as quickly and with as much nuance as I had hoped. This has the makings of a useful tool.
CHAPTER 5. CONCLUSION

5.2 Future Work

5.2.1 Feature engineering from first principles

SketchEngine gramrels were a convenient place to start, but they are not the place to finish this work. The full range of features relevant to the semantic context of a word is a superset of things that can be expressed as gramrels or as syntactic dependencies. Generalizations that are useful at profiling the behavior of single words may be different from those that profile the behavior of synsets. A dependency-parsed corpus is a good foundation for next steps, but it, too, is not enough.

Semantic generalization for nominals and others

I used Roget to generalize about adverbs, and it was not overwhelmingly quirky, but still far from faultless. There are easy improvements to be made in the mapping of words to Roget classes (e.g. limit to the proper part of speech, rather than relying on string matching, to avoid \textit{still.adv} $\rightarrow$ \textit{vaporizer.n}).

The gift of heterogeneity

One happy side-effect of the heterogeneity of ontologies may be exactly the ways in which they differ. The present thesis uses only Roget for simplicity, but there is likely to be some ontological payoff to exploring the consequences of different classification schemes. Any other resource — BabelNet [Navigli and Ponzetto, 2012] and the Historical Thesaurus of the OED [Kay \textit{et al.}, 2009] are two that I look forward to exploiting in future research — would yield different generalizations alone, but especially in concert.

I suspect there is also rich potential in automatically comparing a large number of ontologies for latent semantic knowledge. It could be very interesting to evaluate and characterize
the nature of conflicts between different classificatory hierarchies. While there is reassuring confirmation in their isomorphisms, there may be useful data to be found in their heteromorphisms.

**Non-ontological generalization**

Ontological generalization is not the only necessary path to semasiological generalization. I had thought that CoreLex [Buitelaar, 1998] might have interesting results as a source of generalizations for nominals, but after seeing what I saw with subjects and objects, I believe that it, or something like it, is essential to properly characterize the commonalities of nominal arguments to verbs, without really characterizing other aspects of the corpus from which something is drawn.

**Syntactic generalizations for all parts of speech**

The ‘usage patterns’ from the CLAWS sketch grammar are at best suggestive at this point. I have not spent enough time really grokking what each one means and how it might map to questions of semantic factors. And 26 unary syntactic patterns of is so few that it may not be appropriate to use unsupervised clustering methods to sort through them. And they are only defined for verbs, but Apresjan suggests that other parts of speech may be equally prone to grammatical peculiarities that apply only to a few synonyms in a set.

**5.2.2 Multi-stage clustering**

In many places I encountered confounding collocations which might appear to be ‘artifacts’, but are really evidence of polysemy which must be engineered for and addressed directly. If contexts can be usefully clustered first by ‘word sense’ — ideally some distributional proxy for word sense, since I, too, don’t believe in word senses [Kilgarriff, 1997] — and treated as
separate elements to be clustered, much more specific semantic generalizations at a sense level may be tractable. Which factors are expressed with which senses of polysemous words? A ‘lingering glance’ has different interior motivation than a ‘disapproving glance’ even if both come from a place of emotional unease.

In multi-stage clustering, we may be unsurprised if we found that different senses of words have different collocational tendencies: that certain classes of grammatical relations only happen in concert with other relations. A single-stage clustering would find the probabilities that convince, persuade, encourage, force etc. are concerned with actions, dismissive of the will of the patient, etc.; multi-stage clustering might find that the probabilities of certain relations are conditionally dependent on the presence of other relations — while others, appearing independently, may be taken as ‘core’ contextual cues. This might simply recapitulate distinctions between complements and adjuncts, but it may instead reveal patterns of semantic composition that are yet unknown.

5.2.3 Quantifying probabilities

I haven’t shown many numbers in my rank-ordered tables, but the statistical origins of semantic factors extracted in this way may have uses beyond lexicographical descriptions of meanings. Because they are based on empirical co-occurrence counts in contrast with synonymous words, these synonym descriptions can also support quantification of probabilities:

- Probability that a given lexical unit X implicates semantic factor Y;
- Probability that any member of a given synset implicates semantic factor Y;
- Probability that a given lexical unit X has been chosen specifically to cancel the implicature of some semantic factor
CHAPTER 5. CONCLUSION

Figure 5.1: 2-D clustering of handpicked subset of meanings; edges show covariance

In concerts with the semantic judgments themselves, these probabilities could support a rich range of computations of expressive meaning in context.

5.2.4 Visualization

In figure 2.5 we showed an hand-built schema of plesionym relations for ‘humor’ words. Our clusters can also be visualized in two dimensions, with edges drawn between them to show covariance of shared features. Figs. 5.1 and 5.2 show this for both a large and a small synonym set. These require just as much interpretation as the labeled text clusters, and are not very informative in their current state. But doubtless there is also a path to making them informative and perhaps interactive — leading to a tool that is more useful than the tables that make up the bulk of chapter 4.
Figure 5.2: 2-D clustering of 60 distributionally-similar words; edges show covariance
5.2.5 Composing meanings

The ‘unmarked’ core-meanings words — Pottier’s archilexemes — can perhaps be automatically detected by selecting from the lexical items that are most promiscuously distributed: i.e. modified with the most semantic classes of adverbials, or Agented by the greatest range of subjects. The same distributional patterns should also allow a path to composing meanings: ‘ask nicely’ → request; ‘ask bluntly’ → demand. The current framework is based too heavily around single-token lexical items for this to be tractable, but future work can take it into account with things like skipgrams, pre-processing for common collocations, or other approaches to fuzziness. Such approaches also have obvious utility for multi-word expressions and phraseology.

5.2.6 The value of subcorpus features

In corpora with more complex metadata, the metadata fields can also be features. [Lau et al., 2014] and [Perek, 2014] detected sense divisions through collocational behavior in tagged and historical corpora; the transpose distributional hypothesis can likely also be applied to this sort of question in a polysemous space.

5.2.7 Theoretical challenges remaining

Factives and counterfactives may be hard to recognize this way, without some other annotation: ‘Amina forgot to buy a patch kit’ vs. ‘Amina forgot that she bought a patch kit’: we may recognize the syntactic patterns, but it is not clear that the factivity would be recognizable. Many other semantic tendencies may be equally difficult to detect. But feature engineering in the interest of discovering semantic factors seems worth pursuing for a great many questions, large and small.
CHAPTER 5. CONCLUSION

5.2.8 The reek of genres

What is to be made of the unmistakably journalistic newspaperese that was detectable in the subject/object relations? Is it merely an artifact of the composition of the BNC, or a happenstance of some cases of formulaic language overwhelming a diverse core? Or do nominal arguments have a markedly different relationship to verb semantics than other syntactic dependencies? This question can be addressed, in part, by further exploring nominals with the various approaches I have proposed. But once addressed, this may in turn lead to more questions. So it goes.

5.3 Finally

I believe I have prototyped a novel and useful tool for lexical analysis. At times I have had inklings that my lexicographer friends fear that work like mine will automate their jobs into obsolescence. At this stage, at least, I do not believe that there is a risk of that. The clusters I have made require significant native sprachgefühl to be interpreted properly. This does not mean that there would be no change to the work of a lexicographer under this framework, nor simply a slotting in of synonym-clusters for word sketches. This thesis is called ‘lexicography as feature engineering’ because I suspect future lexicographers may need to engage in iterative exploration of feature space for words as they define them for computational applications. With the right set of features, a definition may almost write itself — but not without human guidance.
Bibliography


BIBLIOGRAPHY


[Roget, 1852] Peter Mark Roget. *Thesaurus of English Words and Phrases Classified and Arranged so to Assist in Literary Composition*. 1852.


