Detecting selectional behavior of complex types in text

Anna RUMSHISKY
Dept. of Computer Science
Brandeis University
Waltham, MA USA
arum@cs.brandeis.edu

Victor A. GRINBERG
Dept. of Computer Science
Northeastern University
Boston, MA USA
victor@ccs.neu.edu

James PUSTEJOVSKY
Dept. of Computer Science
Brandeis University
Waltham, MA USA
jamesp@cs.brandeis.edu

Abstract

In this paper, we discuss some aspects of selectional behavior of dot objects, and present an algorithm for clustering selector contexts for dot nominals according to the selected type. The clustering algorithm is based on the notion of contextualized similarity between selector contexts and defines a similarity measure for contextual equivalents of the target nominal.

1 Introduction

In the Generative Lexicon (GL) (Pustejovsky, 1995) knowledge representation framework, complex types (dot objects) are introduced to account for certain types of inherent polysemy. In this paper, we discuss some aspects of selectional behavior of dot objects in corpus and present a method for automatic detection of selector contexts specific to the component types of the dot. We begin by examining some of the relevant data. We then present an algorithm for clustering selector contexts for dot nominals according to the selected type. We conclude with some preliminary results.

2 GL Background & Data Analysis

Complex types are introduced in GL as a mechanism for dealing with selectional behavior of nouns such as *lunch* (EVENT · FOOD) and *newspaper* ((PHYS · INFO) · ORGANIZATION). The contexts in which complex types occur may select for any of the simple types that make up the complex type.

(1) a. I have my *lunch* in the backpack. (FOOD)
b. Your *lunch* today was longer than usual. (EVENT)

For a dot nominal, the senses that correspond to the simple types are connected in a regular and well-defined manner. Some examples of complex types are given in Table 1. Complex types typically allow multiple selection:

(2) We had a *delicious* (FOOD) *leisurely* (EVENT) *lunch*.

There also exist contexts that select specifically for the complex type of each kind. Thus, for some of the complex types there also seem to exist gating predicates (Pustejovsky, 2007) whose selectional specification may specify a transition between two simple types that make up the complex type. For example, food preparation predicates (e.g. *poach, steam, braise, cook*) are gating predicates for such complex types as ANIMAL · FOOD:

(3) She wouldn't *poach* a *chicken* any other way.

Since some predicates select specifically for complex types, some dot objects may function as disambiguators for such predicates. Consider the verb *dictate*, which has two main senses: (1) "verbalize to be recorded", and (2) "control" (possibly split into "control" with animate subjects and "serve as motivation for" with inanimate subjects). The following nouns all occur[1] as direct objects with the first sense of *dictate*:

(4) a. passage, story, letter, memoirs, novel
b. message, words, work, point

However, the nouns in (4a) are the "good" disambiguators (i.e. they can not be dictated in the "control" sense). The nouns in (4b) are ambiguous. The good disambiguators are actually dot objects of type INFO · PHYSOBJ, with *dictate* functioning as a gating predicate, which requires for the information to be given physical form.

The use of complex types in text suggests that there is an inherent asymmetry in the way dot objects are used. This asymmetry is consistent with the systematic relation between the senses, where each sense corresponds to one of the component types. For example, for the ANIMAL · FOOD nominals, the subject position tends to disprefer the FOOD sense, whereas in the object position, such nominals occur both with the FOOD- and the ANIMAL-selecting predicates, as well as with the gating predicates. In the object position, the FOOD selectors and the gating predicates tend to dominate:

(5) *chicken.n*
subject
a. ANIMAL: peck, look, wander, come, cross, follow, die
object
a. ANIMAL: count, chase, kill, shoot, slaughter, skin, pluck, sacrifice, throw
b. FOOD: eat, serve, prefer, turn, dip, stuff, carve, baste, roast, simmer
c. ANIMAL · FOOD: poach, cook

A similar asymmetry can be seen with respect to different argument positions for such dot types as PROCESS · RESULT, EVENT · PROPOSITION, etc. For example, adjectival modifiers for *construction* (PROCESS · RESULT) tend to select for RESULT, whereas the predicates that take *construction* as direct object tend to select for PROCESS. Similarly, for *allegation* (EVENT · PROPOSITION), the PROPOSITION interpretation is preferred in the object position.

(6) *construction.n*
object
EVENT: finance, oversee, complete, supervise, halt, permit, recommend enable, delay, stimulate
PHYSOBJ: examine, build, inaugurate, photograph
adjectival modifier
PHYSOBJ: logical, syntactic, passive, solid, all-metal, geometric, hybrid, rugged, sturdy, artificial, cultural, imaginative

[1]The data below is taken from the British National Corpus (BNC)

Dot type	Example
ACTION · PROPOSITION	promise, allegation, lie, charge
STATE · PROPOSITION	belief
ATTRIBUTE · VALUE	temperature, weight, height, tension, strength
EVENT · INFO	lecture, play, seminar, exam, quiz, test
EVENT · (INFO · SOUND)	concert, sonata, symphony, song
EVENT · PHYSOBJ	lunch, breakfast, dinner, tea
INFO · PHYSOBJ	article, book, CD, DVD, dictionary, diary, email, essay, letter, novel, paper
ORGANIZATION · (INFO · PHYSOBJ)	newspaper, magazine, journal
ORGANIZATION · LOC · HUMANGROUP	university, city
EVENT · LOCATION · HUMANGROUP	class
APERTURE · PHYSOBJ	door, window
PROCESS · RESULT	construction, imitation, portrayal, reference, decoration, display documentation, drawing, enclosure, entry, instruction, invention, simulation, illustration, agreement, approval, recognition, damage, compensation, contribution, discount, donation, acquisition, deduction, endowment, classification, purchase
PRODUCER · PRODUCT	Honda, IBM, BMW
TREE · FRUIT / TREE · WOOD	apple, orange, coffee / oak, elm, pine
ANIMAL · FOOD	anchovy, catfish, chicken, eel, herring, lamb, octopus, rabbit, squid, trout
CONTAINER · CONTENTS	bottle, bucket, carton, crate, cup, flask, keg, pot, spoon

Table 1: Some examples of dot objects of different complex types, as well as "pseudo-dots" that exhibit dot-like behavior due to coercion.

(7) *allegation.n*
object
EVENT: face, fuel, avoid, deflect
PROPOSITION: deny, refute, counter, contain, substantiate, rebut, confirm, believe, corroborate, hear, dispute, broadcast, prove

Generic asymmetry of use (i.e. the asymmetry across all argument positions) is also a common property of some dot nominals. For example, such PROCESS · RESULT nominals as *building*, *invention*, *acquisition* show a distinct preference for one of the types in all argument positions. For *building* and *invention*, the RESULT/PHYSOBJ interpretation is much more frequent, whereas for *acquisition*, the PROCESS/EVENT interpretation dominates the use in all argument positions. In (8), (9), and (10) below, we list the lexical items that tend to select each component type (or the dot type itself) for these nouns in selected argument positions[2].

(8) *invention.n*
object
a. RESULT: produce, explain, protect, adopt, develop, combine, patent, license, display, neglect, export, exploit
b. PROCESS: welcome, avoid, stimulate, spark, trace, facilitate, demand

[2]Note that for *building*, for example, *plan* selects for the complex type EVENT · RESULT in the object position, while *abandon* may select for either of the component types.

subject
a. RESULT: simplify, impress, consist, popularize, appear, comprise
adjectival modifier
a. RESULT: finest, original, comic, successful, British, latest, patented, brilliant

(9) *building.n*
object
a. PHYSOBJ: erect, demolish, construct, occupy, restore, enter, convert, design, destroy, lease, own, renovate, surround, damage, complete
b. EVENT: allow, finish, oppose, accelerate, initiate, halt, commence, stop, undertake
c. EVENT · RESULT: plan
d. EVENT, RESULT: arrange, abandon
subject
a. PHYSOBJ: house, stand, collapse, contain, survive, belong, remain, overlook, surround, fall, replace, dominate
b. EVENT: begin, continue, commence
c. EVENT · PHYSOBJ: date
d. EVENT, PHYSOBJ: accompany

(10) *acquisition.n*
object
a. EVENT: finance, fund, complete, announce, authorize, commence, facilitate, oversee, control, approve, undertake
b. RESULT: identify, secure, seize, store, stalk
subject
a. EVENT: occur, boost, result, strengthen, increase, depend, form, take, continue, affect, result
b. RESULT: turn out, offer, comprise, bore, allow
c. EVENT · RESULT: put, increase, mean, represent, complement

Subphrasal syntactic cues (e.g. plural/singular, definite/indefinite article) are often strong indicators of the likely type selection:

(11) a. He stored all his new *acquisitions* here. (**plural**, RESULT)
b. The city authorized the *acquisition* of land to build the tunnel. (**singular**, EVENT)

(12) a. It was the most important development in radio since *the invention* of the transistor. (**definite**, EVENT)
b. *An invention* may be very beneficial, but it might also seriously undermine an existing business. (**indefinite**, RESULT)

However, the asymmetry inherent in a particular dot object may easily overrule even the strong contextual indicators. For example, *acquisition* still tends to favor the EVENT interpretation even in plural, whereas even the use with an aspectual predicate does not override the preference of *building* for the RESULT interpretation:

(13) a. *Acquisitions* have formed an important part of our strategy.
b. The *building* was never *completed*.

3 Clustering Task

This complexity of selectional behavior makes it difficult to apply to dot objects the notion of word sense as it is used in various automatic text processing tasks. For example, multiple selection (cf. (2)) makes it impossible to resolve the classification problem of word sense disambiguation. However, as illustrated in (9), (10), (7), (11), and (12), in many cases, it is possible to tell which type (or types) a particular individual selector prefers. In this work, we address this task. Our goal in these experiments is to obtain a clustering of all selectors (headwords for all grammatical relations a dot object is found in) according to the type it selects from the complex type. Hence, for *lunch*, we would like to obtain groupings such as:

(14) *lunch.n*
object
a. FOOD: eat, cook, enjoy, prepare, take, bring, etc.
b. EVENT: skip, finish, attend, miss, host, cancel, etc.
adjectival_modifier
a. FOOD: light, delicious, three-course, excellent, liquid, home-cooked, half-eaten, heavy, substantial, etc.
b. EVENT: leisurely, early, annual, celebratory, official, private, weekly, etc.

To address this problem, we developed a clustering method based on *contextualized similarity*. We define *contextualized similarity* as similarity between two lexical items with respect to a particular context. In this work, context is defined as a single populated syntactic relation, in line with the way context is typically defined in the distributional similarity literature (Grefenstette, 1994; Lin, 1998; Dagan, 2000; Pantel and Lin, 2002). For example, *cook* and *prepare* both occur in the context (*lunch*, **object**$^{-1}$) with a certain frequency.[3]

Whereas two lexical items may not be distributionally similar overall, in a particular context they may be essentially equivalent. This equivalence is in terms of the aspect of meaning they select. For example, *cancel* and *attend* each have very different sets of senses, and their frequencies of occurrence do not have a similar distribution across contexts. However, with respect to the context (*lunch*, **object**$^{-1}$), they are quite similar: they both select for the EVENT interpretation. We use the notion of *contextual equivalence* to capture this intuition. A lexical item w_1 is a *contextual equivalent* of lexical item w_2 with respect to a certain grammatical relation R if one of its senses selects for the same aspect of meaning as one of the senses of w_2 in the argument position defined by R.

We use the following idea. Consider a bipartite graph where one set of vertices corresponds to headwords and the other to dependents, under a relation R. Each relation can be viewed as a function mapping from headwords to dependents. The relation is defined by a set of tuples (w, R, w'), where w is the head, and w' is the dependent. The inverse of each relation is then a set of tuples (w', R^{-1}, w).

Clustering selector contexts for the target word according to the type they select (e.g. predicates that select for the EVENT interpretation of *lunch* vs. those that select for the FOOD interpretation) can thus be induced by clustering *contextual equivalents* of the target word - and vice versa.[4]

3.1 Contextualized Similarity

In the experiments described below, we apply the contextualized similarity metric to the contextual equivalents of the target word. We proceed as follows:

1. Identify the set of selector contexts in which the target word was found in corpus. For the target context $(t, R) =$ (*lunch*, **object**$^{-1}$), this gives a set of verbs such as those listed in (14) above.

2. Take the inverse image of the above set under the R^{-1} relation (in this case, **object**), which gives a set of nouns which occur with selectors of the target word. These are candidates for contextual equivalence for different senses of the target word (cf. Fig 1). A noun is considered a potential contextual equivalent only if it occurs in relation R with the specified number (or percentage) of the target's selectors. We used the threshold $\sigma = 5$. These are the elements selected for primary clustering.

[3]**Object**$^{-1}$ is the inverse of the **object** relation that holds between *prepare* and *lunch*.
[4]This graph representation is similar the one used in literature more commonly for symmetric relations such as conjunction or apposition (Widdows and Dorow, 2002).

3. For every word in the set of candidates for contextual equivalency, we obtain a set of "good" selectors:

   (a) Take all the selector contexts in which both the target and the contextual equivalent are found. Compute two conditional probability scores for each selector s: $P(s|Rn)$ and $P(s|Rt)$, where s is the selector context, n is the potential contextual equivalent, and t is the target word. Notice that selectors are verbs for the **object** relation, adjectives for the **a_modifier** relation, and so on.

   (b) Identify the "good" selectors, i.e. those that select the same interpretation both for the target noun and for potential contextual equivalent. For example, for *sandwich*, given the target word *lunch*, we would need to select verbs such as *eat, cook, make,* etc. The "good" selectors will have relatively high conditional probabilities with both words. It's important to understand that the conditional probability will depend on how frequent the appropriate sense is for each of the two words. In the experiments below, we used the geometric mean to pick the "good" selectors. We compute the geometric mean of the above conditional probabilities, and choose the top-K selectors that maximize it. In the present experiments, we used $K = 20$.

4. Compute the similarity matrix for the potential contextual equivalents. We compute the similarity measure as the sum of minima, which is effectively equivalent to set-theoretic overlap used in Jaccard and Dice measures. The contextualized similarity for two potential contextual equivalents w_1 and w_2 is computed as the sum of minima of conditional probabilities for every "good" selector in the list obtained for w_1 and w_2:

$$csim_K(w_1, w_2, (t, R)) = \sum_{s \in S^K_{w_1,t} \cap S^K_{w_2,t}} \min\left(P(s|Rw_1), P(s|Rw_2)\right)$$

   where t is the target word, R is the grammatical relation, and $S^K_{w,t}$ are the sets of top-K good selectors that pick the same sense of w and t.

   Unlike the standard numerical extensions of Jaccard and Dice, we do not normalize the sum of minima either by the size of the union, or by the average size of each set S_i in order to avoid high similarity scores for high-frequency words among potential contextual equivalents. These are effectively promiscuous collocates that occur frequently with all selectors, including the "good" (i.e. reliable) selectors for each of target word's senses. The conditional probabilities for them, however, are low due to their high frequencies. Normalizing the sum of minima by the sum of maxima, for example, as in Jaccard, will bring the similarity value up for high-frequency pairs: both words in the pair will have roughly equally low conditional probability for all verbs in their respective selector lists.

5. Perform agglomerative hierarchical clustering. We experimented with the contextualized similarity metric using both group-average and cluster centroid methods. The results reported here were obtained using group-average clustering.

6. Compute Average Pairwise Similarity (APS) between the elements of each cluster. As we proceed from the bottom of the dendrogram up, APS for the clusters decreases. We compute the percent decrease in APS (APS derivative) for every cluster merge point. The dendrogram will be cut at the points that have high percent decrease in APS, so as to select the clusters obtained prior to the APS-decreasing merge, as specified below.

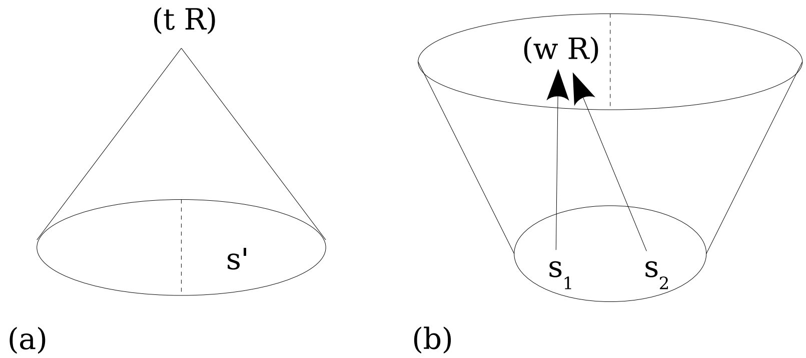

Figure 1: Clustering contextual equivalents with inverse image

7. Select a certain number of *seed* elements from the contextual equivalents with highest contextualized similarity to the target. Trace their merges in the dendrogram, obtaining a trace sequence of clusters $C_0^i \subset ... \subset C_{n_i}^i$ for each seed i, where $C_0^i = \{i\}$, and $C_{n_i}^i$ is always the top cluster.

   (a) Sort the clusters in each trace sequence on the percent decrease in APS obtained at the next merge, and select the top-scorers.

   (b) If a cluster is among the top-scorers for several seeds, select that cluster to represent one of the senses of the target.

8. For each of the target's selectors s in grammatical relation R, compute the following score for each of the chosen clusters C:

$$assoc(s, C) = \sum_{w \in C} P(s|Rw)$$

   The resulting score indicates how likely selector s is to pick the sense of the target associated with C. The difference between the scores obtained for different senses with a given selector indicates how strongly that selector tends to prefer one of the senses. If the difference is small, the selector must either (1) select for the complex type itself, or (2) equally likely select for either of the component types.

We used the Sketch Engine library (Kilgarriff et al., 2004) to extract and access populated grammatical relations from the British National Corpus. The Sketch Engine parser extracts grammatical relations using regular expressions over pos-tagged text. A number of patterns is defined for each relation, so the **subject** relation, for example, is extracted in both active and passive. Most of the extracted binary relations, such as **object/object_of**, **subject/subject_of**, **a_modifier/modifies**, etc. are bidirectional.

In the current preliminary experiments, we ran the clustering procedure for the following dot object/argument positions pairs: (*allegation.n*, **object_of**), (*building.n*, **object_of**), (*chicken.n*, **object_of**), (*lecture.n*, **object_of**), (*lecture.n*, **a_modifier**), (*lunch.n*, **object_of**), (*lunch.n*, **a_modifier**), (*newspaper.n*, **object_of**), (*newspaper.n*, **a_modifier**).

3.2 Results

Evaluation of unsupervised distributional clustering algorithms is typically done by comparing the results to manually constructed resources such as WordNet, Roget's thesaurus, MRD definitions, etc. (Grefenstette, 1994; Lin, 1998; Pantel and Lin, 2002; Widdows and Dorow, 2002). Thesaurus entries against which the results are compared are typically cross-checked against multiple resources or frequency-filtered using semantically tagged corpora (Pantel and Lin, 2002). This method of evaluation is not suitable for the task of clustering selectors, since the grouping obtained is only valid with respect to the specified target context. Using resulting clusters in a standard task such as word sense disambiguation is also problematic, since sense-tagged corpora typically assign a single sense to one token, which makes it impossible to account for multiple selection. We used manual inspection of selected clusters to get an idea about the validity of obtained clusters. We illustrate below the outcome of different stages of the algorithm using the target context $tR =$ (*lunch*, **object**$^{-1}$).

Consider the best clusters obtained for FOOD- and EVENT-contextual equivalents for *lunch* used in direct object position:

```
Cluster 6290=5702+6230:
[juice-n,cocktail-n,alcohol-n,wine-n,ale-n,brandy-n,vodka-n,champagne-n,beer-n,pint-n,
whisky-n,gin-n,sherry-n,straw-n,corn-n,liver-n,cereal-n,goose-n,vegetable-n,rice-n,
pasta-n,stuffing-n,dish-n,tomato-n,pea-n,bean-n,ham-n,turkey-n,mushroom-n,potato-n,
chicken-n,carrot-n,bacon-n,cabbage-n,nut-n,apple-n,orange-n,lettuce-n,dessert-n,chip-n,
food-n,snack-n,buffet-n,steak-n,salad-n,sandwich-n,dinner-n,meal-n,lunch-n,breakfast-n,
supper-n,beef-n,sweet-n,crisp-n,chop-n,sausage-n,pizza-n,meat-n,chocolate-n,banana-n,
spaghetti-n,yoghurt-n,ice-cream-n,doughnut-n,mint-n,honey-n,jam-n,soup-n,toast-n,tea-n,
coffee-n,bread-n,cheese-n,cake-n,curry-n,bun-n,biscuit-n,pudding-n,marmalade-n,jelly-n,
pie-n,porridge-n,tart-n,pastry-n,stew-n,sauce-n,hay-n,butter-n,roll-n,cream-n]

Cluster 6347=5673+6299:
[tournament-n,contest-n,outing-n,barbecue-n,exhibition-n,festival-n,hearing-n,summit-n,
talk-n,ballot-n,election-n,referendum-n,disco-n,congress-n,inquest-n,fair-n,ceremony-n,
reunion-n,rally-n,meeting-n,conference-n,seminar-n,parade-n,rehearsal-n,dance-n,funeral-n,
clinic-n,feast-n,celebration-n,session-n,workshop-n,demonstration-n,concert-n,briefing-n,
lecture-n,reception-n,banquet-n,luncheon-n,wedding-n,gathering-n,event-n,procession-n]
```

Table 2 shows the soft selector assignment obtained for (*lunch*, **object**$^{-1}$) using the above clusters as described in step 8 in Section 3.1. Notice that the selector sets for both senses are quite heterogeneous. Nonetheless, the assigned selector/sense pairings are correct with accuracy well above the majority baseline. For example, out of all selectors, *hold* gets assigned the highest association score with the EVENT sense. This may appear inaccurate, since *hold* is quite polysemous and one of its senses selects for PHYSOBJ. However, in all occurrences of *lunch* in the BNC, *hold* is indeed found with the EVENT interpretation, actually confirming the accuracy of the assigned scoring.

A partial trace of the dendrogram obtained for (*lunch*, **object**$^{-1}$), using the seed *conference* is shown in Table 4. It is easy to see that semantically very distinct words begin to cluster very early in the trace, yet most of the elements in the initial merges are clearly good contextual equivalents for the EVENT sense of *lunch*. Table 3 illustrates the choice of selector lists based on which contextualized similarity is computed (cf. steps 3 and 4 in Section 3.1). The accuracy of the resulting selector/sense assignment clearly depends on the success at each stage of the algorithm. The current implementation seems to produce selector lists that are more accurate for verb-object selectors than for adjectival modifiers.

Selector	FOOD	EVENT	Assignment	Selector	FOOD	EVENT	Assignment
eat-v	0.089	0.002	food	cancel-v	0.000	0.003	event
cook-v	0.024	0.003	food	organise-v	0.000	0.034	event
serve-v	0.024	0.002	food	include-v	0.013	0.011	food
skip-v	0.002	0.000	food	order-v	0.008	0.001	food
finish-v	0.009	0.002	food	grab-v	0.000	0.000	food
enjoy-v	0.006	0.016	event	give-v	0.010	0.045	event
prepare-v	0.009	0.004	food	spoil-v	0.000	0.000	food
attend-v	0.001	0.100	event	share-v	0.004	0.002	food
miss-v	0.001	0.002	event	hold-v	0.004	0.157	event
take-v	0.023	0.007	food	pack-v	0.000	0.000	food
provide-v	0.007	0.010	event	appreciate-v	0.000	0.000	food
get-v	0.064	0.014	food	like-v	0.032	0.004	food
bring-v	0.011	0.003	food	offer-v	0.006	0.003	food
buy-v	0.023	0.000	food	plan-v	0.000	0.013	event
arrange-v	0.002	0.019	event	supply-v	0.001	0.000	food
want-v	0.035	0.003	food	make-v	0.083	0.016	food
host-v	0.000	0.010	event	organize-v	0.000	0.011	event

Table 2: Selector assignment scores for (*lunch*, **object**$^{-1}$)

	lunch-n	conference-n	fair-n		lunch-n	conference-n	fair-n
attend-v	0.020	0.125	0.066	arrange-v	0.011	0.002	0.011
hold-v	0.013	0.180	0.264	host-v	0.005	0.016	0.033
tell-v	0.003	0.136	0.022	follow-v	0.007	0.009	0.011
organise-v	0.008	0.038	0.011	organize-v	0.003	0.013	0.011

Table 3: Conditional probabilities $P(s|Rn)$ for the overlap in top-K selector lists for *conference* and *fair*, with respect to (*lunch*, **object**$^{-1}$)

4 Conclusions and Future Work

Despite the peculiar selectional behavior of dot objects, which includes occurring in multiple selection contexts as well as with dot-type-specific selectors, we have shown that it is possible to derive automatically sets of selectors for each of the component types using the method we described. The best clusters of contextual equivalents obtained for each argument position in which the target nominal occurs clearly correspond to the component types of the dot. Examining the difference between the association scores obtained for each selector with respect to the resulting clusters seems to produce heterogeneous sets of "good" disambiguators for each component type.

We avoid the common computational pitfalls in distributional similarity-based clustering by computing clusters of short contextualized vectors. A bipartite relation graph allows for mirror clustering of the target's selectors and its contextual equivalents. It is also quite easy to consider extended relation sets instead of single relation inverses. This is clearly desirable in a standard WSD task, where a particular aspect of meaning gets picked by a combination of selectors in different argument positions. Alternatively, it is also possible to combine the association scores from selectors in different argument positions to disambiguate the target.

The clustering results may potentially be improved by making it into an iterative procedure, repeated in succession on both parts of the graph. This procedure also allows one to seed the clusters

Step	*Inter-cluster APS*	*Intra-cluster APS*	*APS % decrease*	*Resulting cluster*
1	0.445	0.445	0.00	[**conference-n**] [seminar-n]
2	0.430	0.435	0.02	[meeting-n] [**conference-n** seminar-n]
3	0.397	0.416	0.04	[rally-n] [meeting-n **conference-n** seminar-n]
4	0.342	0.387	0.07	[reunion-n] [rally-n meeting-n **conference-n** seminar-n]
5	0.314	0.363	0.06	[ceremony-n] [reunion-n rally-n meeting-n **conference-n** seminar-n]
6	0.295	0.332	0.09	[inquest-n fair-n] [ceremony-n reunion-n rally-n meeting-n **conference-n** seminar-n]
7	0.267	0.318	0.04	[congress-n] [inquest-n fair-n ceremony-n reunion-n rally-n meeting-n **conference-n** seminar-n]
8	0.264	0.307	0.03	[disco-n] [congress-n inquest-n fair-n ceremony-n reunion-n rally-n meeting-n **conference-n** seminar-n]
9	0.246	0.280	0.09	[talk-n ballot-n election-n referendum-n] [disco-n congress-n inquest-n fair-n ceremony-n reunion-n rally-n meeting-n **conference-n** seminar-n]
10	0.223	0.272	0.03	[summit-n] [talk-n ballot-n election-n referendum-n disco-n congress-n inquest-n fair-n ceremony-n reunion-n rally-n meeting-n **conference-n** seminar-n]
11	0.216	0.265	0.03	[hearing-n] [summit-n talk-n ballot-n election-n referendum-n disco-n congress-n inquest-n fair-n ceremony-n reunion-n rally-n meeting-n **conference-n** seminar-n]
12	0.197	0.224	0.15	[hearing-n summit-n talk-n ballot-n election-n referendum-n disco-n congress-n inquest-n fair-n ceremony-n reunion-n rally-n meeting-n **conference-n** seminar-n] [parade-n rehearsal-n wedding-n funeral-n clinic-n feast-n celebration-n session-n workshop-n demonstration-n concert-n briefing-n lecture-n reception-n banquet-n luncheon-n]
...				...

Table 4: Dendrogram trace for the target `lunch-n`, seed `conference-n`.

manually, as it is done in some thesaurus construction algorithms (e.g. (Roark and Charniak, 1998)).

References

I. Dagan. 2000. Contextual word similarity. In R. Dale, H. Moisl, and H. Somers, editors, *Handbook of Natural Language Processing*. Marcel Dekker.

G. Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, Norwell, MA, USA.

A. Kilgarriff, P. Rychly, P. Smrz, and D. Tugwell. 2004. The Sketch Engine. *Proceedings of Euralex, Lorient, France*, pages 105–116.

D. Lin. 1998. Automatic retrieval and clustering of similar words. *COLING-ACL, Montreal, Canada*.

P. Pantel and D. Lin. 2002. Discovering word senses from text. In *Proceedings of ACM SIGKDD02*.

Patrick Pantel. 2003. *Clustering by Committee*. PhD dissertation, Department of Computing Science, University of Alberta.

Fernando Pereira, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of english words. In *Meeting of the Association for Computational Linguistics*, pages 183–190.

J. Pustejovsky. 1995. *Generative Lexicon*. Cambridge (Mass.): MIT Press.

J. Pustejovsky. 2007. Type Theory and Lexical Decomposition. In P. Bouillon and C. Lee, editors, *Trends in Generative Lexicon Theory*. Kluwer Publishers (in press).

B. Roark and E. Charniak. 1998. Noun-phrase co-occurrence statistics for semi-automatic semantic lexicon construction. page 1110.

D. Widdows and B. Dorow. 2002. A graph model for unsupervised lexical acquisition. In *19th International Conference on Computational Linguistics*, pages 1093–1099, Taipei, Taiwan, August.