Automatic Fine-Grained Semantic Classification for Domain Adaptation

Maria Liakata
University of Wales, Aberystwyth (UK)
email: mal@aber.ac.uk

Stephen Pulman
University of Oxford (UK)
email: sgp@clg.ox.ac.uk

Abstract

Assigning arguments of verbs to different semantic classes ('semantic typing'), or alternatively, checking the 'selectional restrictions' of predicates, is a fundamental component of many natural language processing tasks. However, a common experience has been that general purpose semantic classes, such as those encoded in resources like WordNet, or handcrafted subject-specific ontologies, are seldom quite right when it comes to analysing texts from a particular domain. In this paper we describe a method of automatically deriving fine-grained, domain-specific semantic classes of arguments while simultaneously clustering verbs into semantically meaningful groups: the first step in verb sense induction. We show that in a small pilot study on new examples from the same domain we are able to achieve almost perfect recall and reasonably high precision in the semantic typing of verb arguments in these texts.

1 Introduction

Since the earliest days of computational linguistics the semantic properties of verbal arguments have played an important role in processing. Many classic types of ambiguity, and hence their resolution, depend on this: 'flying planes can be dangerous' is ambiguous because 'flying planes' can describe an activity or a plural entity, either of which can be a semantically appropriate subject of 'be dangerous', whereas 'swallowing apples can be dangerous' does not display this ambiguity. Both 'fly' and 'swallow' can be transitive or intransitive, but whereas 'planes' is both a semantically appropriate subject for intransitive 'fly' and an appropriate object for transitive 'fly', 'apples' is not a semantically appropiate subject for intransitive 'swallow'. Semantic (mis)typing rules out this syntactically valid combination. Similarly, an important component of reference resolution is the knowledge of what semantic category an entity falls under. For example, in 'The crop can be used to produce ethanol. This can be used to power trucks or cars', knowledge that ethanol is the kind of thing that can be subject of 'power', whereas 'crop' is not, is required to successfully resolve the reference of 'this'.

When considering division into semantic categories one's immediate thought would be to take advantage of existing semantic resources (such as WordNet (Miller, 1995)) or FrameNet (Baker et al., 1998). For example, Clark and Weir (2002) calculate the probability of a noun sense appearing as a particular argument by using WordNet to generalise over the noun sense. However, even though WordNet has been extremely useful in numerous applications, many researchers have found that the fact that it is largely developed via the intuitions of lexicographers, rather than being empirically based, means that the semantic information often is poorly matched with word usage in a particular domain. Pantel and Lin (2002) and Phillips and Riloff (2002) have pointed out that WordNet often includes many rare senses while missing out domain-specific senses and terminology. Some authors, Kilgariff (1997) and Hanks and Pustejovsky (2004), among others, reject the basic idea shared by WordNet and FrameNet (as well as traditional dictionaries) that there is a fixed list of senses for many verbs, arguing that individual senses will often be domain specific and should be discovered empirically by examining the syntactic and semantic contexts they occur in. We are highly sympathetic to this view and in this work we assume, as Hanks and Pustejovsky do, that rather than relying on the intuitions of a lexicographer, it is better to try to induce verb senses and semantic types automatically from data drawn from the domain of interest.

In this paper we report on some experiments in learning semantic classes. We carry out prior syntactic and semantic analysis of a relevant corpus so that verb+argument pairs can be identified. Since we are interested in domain specific semantic classification we make the 'one sense per corpus' hypothesis and ignore word sense disambiguation. For a given verb, we find the head nouns occurring in the subject, object and indirect object noun phrases (where they exist) occurring frequently within the corpus. Now that we have information about nouns co-occurring in different argument slots of verbs we cluster the verbs according to shared argument slots: verbs which have an argument slot (not necessarily the same one) occupied by members of the same cluster are in turn clustered together. The effect of this is to derive noun clusters characterising the semantic types of the argument slots for individual verbs (learning selectional

restrictions) while simultaneously clustering verbs which have similar argument slots. In the case where the same argument slot is involved across verbs, the effect of this is to induce a fine-grained semantic classification of verbs (the first step in learning verb senses). Where different argument slots are involved the effect is to suggest more complex causal or inferential relations between groups of verbs.

To give a simple illustration, if *admit*, *deny*, *suspect* all take the word 'wrongdoing' as their object, then *admit_arg*2, *deny_arg*2, *suspect_arg*2[1] are clustered together into one group *A*. If we also find that words like 'oversight' also appear frequently in the same argument position with roughly the same set of verbs, then 'oversight' will be clustered with the other fillers of group *A*.

A side-effect of the process is a classification of the verbs as well: if *admit_arg*1, *deny_arg*1 and *admit_arg*2, *deny_arg*2 respectively take the same values, 'deny' and 'admit' are clustered together. We may also note that the same classes occur in different argument slots of different verbs: in Liakata and Pulman (2004) we showed how this could lead to the discovery of causal relations specific to a domain: for example (in company succession events), that A succeeds B if B resigns from position C and A is appointed to C.

In the remainder of the paper we describe this clustering process in more detail. We also describe a simple pilot evaluation, by taking two unseen texts from the same domain, and observing to what extent the semantic groupings arrived at can be used to assign semantic types to arguments of verbs. We were pleasantly surprised to find almost perfect recall, and respectable precision figures.

2 Method

The method of clustering together verb argument slots for obtaining domain specific verb senses (either in terms of verb classes or through the assignment of semantic types to the verb arguments) is applied as a proof of concept to the domain of financial news. We chose this domain since the WSJ section of the Penn Treebank II is already available in the form of predicate-argument structures, obtained according to the method described in Liakata and Pulman (2002). However, the same approach can apply to predicate-arg structures from non-treebank data such as the QLFs derived from LFG structures in Cahill et al. (2003) or semantic representations such as in Bos et al. (2004).

The WSJ corpus consists of 2,454 articles with a total of 2,798 distinct verb predicates, 62 prepositional predicates and 221 copular predicates containing the verb to 'be'. Here we are only dealing with the non-prepositional predicates. The latter follow an uneven distribution of occurrences; there is a minority of very frequent verbs whereas the majority are rather sparse. The problem with infrequent predicates is that the number of instances is often too small to allow for meaningful clustering of the verb-argument slots. To circumvent this, we pre-process predicates with low frequencies (freq < 5) by looking them up in WordNet to find the conceptual group (synset) to which they belong and assigning to them the frequency of the member of the synset with the greatest count of occurrences in the corpus. Thus, words featuring as arguments of the most frequent synset member are counted as arguments of the

[1] arg1 is subject; arg2 is direct object; arg3 is indirect object, roughly

less frequent semantically related predicate, so that the latter receives a count boost. For example, the words that appear as subjects of the verb 'hit' are also considered subjects of the verb 'clobber', which belongs to the same synset as 'hit' but is under-represented in the corpus. This is making use of the knowledge that semantically similar verbs are similar in terms of subcategorisation (Korhonen and Preiss, 2003) and is in agreement with the approach in Briscoe and Carroll (1997) where the subcategorisation frames (SCFs) of representative verbs are merged together to form SCFs of the rest of the verbs belonging to the same semantic class. We understand that the above process may be indirectly adding false positives to the verb senses. It would be interesting in the future to examine the trade-off between boosting the counts of infrequent verbs and the addition of false positives.

A second pre-processing stage was applied to the arguments of the 2,798 verb predicates. The idea underlying this process was to create a version of the predicates where obvious semantic grouping would have already taken place. This involved merging together the instances of 'named entity' classes: person names, companies, locations, propositions, money expressions, and percentage and numeric expressions. Company names and suffixes, locations and people's first names are contained in a gazetteer list collected from internet resources.

Since the similarity of argument slots of predicates is to be determined by how many common fillers they share, it is natural to use the Vector Space Model (VSM) originally from Information Retrieval to define similarities. In IR documents containing the same words are considered to be similar. Argument slots of predicates can be characterised by their filler words in the same way that a document is characterised by the words it contains. This means a predicate argument slot can be modelled in terms of a vector of filler-word frequencies.

In order to apply clustering methods to the predicate-arguments, we combined them into a matrix, where each row corresponds to a verb-argument slot (verb-subj/arg1, verb-obj/arg2 or verb-iobj/arg3) and the columns correspond to words-fillers of the verb-argument slots. Each cell 'w_{ij}' in the matrix represents the frequency of word 'j' as a filler of predicate argument slot 'i'. However, even after the first step of grouping together named entities of the same type (as described above) there were 32,990 distinct possible fillers of the three argument positions of the 2,798 verb predicates. By including all possible word fillers as columns, we would end up with a very sparse matrix of $2,798 * 3 = 8,394$ rows and 32,990 columns.

To reduce the size of the matrix it was essential to select a small number of words as representatives of the argument fillers. Even though the literature for feature selection is vast when it comes to supervised machine learning methods, there is very little on feature selection for clustering. Principal Component Analysis (PCA) would be one alternative here as it reduces dimensionality while preserving as much of the variance in the high dimensionality space as possible. However, since PCA does not consider class separability information there is no guarantee that the direction of maximum variance will contain good features for discrimination. In this preliminary experiment we decided simply to use the 100 most frequent words as features. Thus the new matrix is of the order $8,394 * 100$.

2.1 Clustering method

To perform the clustering we chose a probabilistic clustering method which allows instances to belong to more than one class with different probabilities, as this gives a better indication of the quality of each class and agrees more with the intuition that there is more than one possibility for defining the groups. In addition to this, we do not know what the expected number of classes is so ideally we would like the clustering algorithm to predict the optimal number of classes. For the previous reasons we decided to use Autoclass (Cheeseman and Stutz, 1995) which is a system for unsupervised classification, consisting of a classical mixture model enhanced by a Bayesian method for determining the optimal classes.

Autoclass is an extension of the mixture model as each instance can be characterised by multiple attributes instead of just one, so that the dataset is represented as a matrix of attribute values. One need not specify the exact number of clusters since the system first performs a random classification which it then improves through local changes. Autoclass adopts a fully Bayesian approach by assuming a prior probability distribution for each parameter. In the current experiment the instances in each class (verb argument slots) were assumed to follow a log normal distribution.

2.2 The verb argument clusters

Autoclass performed over 500 trials to converge to various solutions with differing numbers of clusters. The most probable clustering, i.e. the one with the highest log posterior, corresponds to 32 classes and was obtained after 200 trials. The high number of classes[2]returned may be due to overfitting of the model or may be a sign that there is not a clear structure in the data itself.

A class obtained from Autoclass is characterized by its class weight, the number of verb-argument slots that constitute its members, as well as its class strength and class cross entropy. There are very heavy, populous classes (e.g. class 0 with 5,571 members) and lightweight, scantily populated classes (e.g. class 31 with 34 members). Class strengths are defined by the mean probability that any instance belonging to a class would have been generated by its probabilistic model. The higher the class strength the more meaningful the class. The strongest class is class 0 followed by class 2, the third most populous class. Admittedly, the class strength for the rest of the classes is very small casting some doubt over the model's predictive power with respect to the data set. Almost every instance is assigned to a class with probability 1, which means that the classes are clearly separated. Class cross entropy, how strongly the model helps differentiate each class from the whole dataset, ranges from zero, for identical distributions, to infinite for distributions that make a complete separation between differing values of the same attribute. A class is more meaningful if its distribution is distinct from the global distribution. In this case class cross entropy has a value of over 118 for every class, suggesting that classes are distinct from the distribution of the data set. Attributes with the most overall influence in classification are the ones corresponding to precise concepts associated with specific contexts such as 'spokeswoman', 'reporter', 'source'. The least useful features for the classification are the ones with the most scattered frequencies across predicates such as 'person'. Thus, one should employ frequent features with counts concentrated in a subset of

[2]From here onwards we shall be using the terms 'class' and 'cluster' interchangeably

predicates and penalise the significance of words with counts distributed evenly in all predicates. This suggests that it would be more reliable to use freq*idf[3] as a criterion for feature selection.

2.3 Interpretation of the clusters

In order to be able to interpret classes, class membership was inspected by processing the class report showing which cases belong to a particular class. Class 0 contains nearly all of the third arguments (indirect objects) of all verbs, which are usually propositions functioning as verb complements. The interpretation of the resulting classes is not straightforward, though about half have some intuitive basis once outliers are ignored. For example, class 9 (below) seems to hold first arguments (subjects) of verbs denoting sudden movement and numeric change such as:

```
class(9,['add_arg1','add_up_arg1',
'back_arg1','balloon_arg1','base_arg2',
'base_arg1','bear_arg1','begin_arg1',
'bestow_arg1','block_arg1','blossom_arg1',
'blow_arg1','blow_up_arg1','come_up_arg1',
'boost_arg2','bother_arg1','break_arg1',
'break_arg2','breathe_arg1','bring_in_arg1',
'bud_arg1','build_up_arg1','bump_up_arg1',
'clean_up_arg1','clear_arg1','climb_arg1',
'come_along_arg1','come_back_arg1','cut_arg2',
'come_down_arg2','come_on_arg1','boom_arg1',
'continue_arg1','contract_arg1','deal_arg1',
'contribute_arg1','count_arg1','count_arg2',
'crack_arg1','crash_arg1','come_down_arg1'
'cut_arg1','cut_down_arg1','contrast_arg2',
'decline_arg1','deduct_arg1','defend_arg2',
'deflate_arg1', 'deflate_arg2','settle_arg1',
'set_off_arg1','shape_up_arg1','shine_arg1',
'shoot_arg1','shorten_arg1','sink_arg1',
'sit_arg1','sit_down_arg1','slip_arg1',
'slip_in_arg1','slump_arg1','soar_arg1']).
```

A closer look at the filler words of the above verbs show that most of them are 'financial indicators' of some sort such as the following:

CLIMB_arg1: share, asset, imports, exports, fund, price, rate, percentage, stock, wages, dollar, dividend, income, volume, market, capital, interest, trading, demand, maker, cost, index, new_bank_index.

SINK_arg1: percentage, yield, stake, wages, stock, index, share, dollar, georgia_gulf_stock, money, company, bank, income, investment, dividend, payout, payroll.

SOAR_arg1: earnings, asset, yield, location, exports, imports, purchase, fund, price, rate, number, wages, share, stock, rating, bid, dollar, interest, dividend, profit, income, volume, risk, holder.

DROP_arg1: borrowing, imports, increase, market, share, investor, surge, capital, money, company, auction, firm, price, bank, limit, scale, holder, profit, dollar, performance, asset, stoc, rate, index, bid, earnings, volatility.

[3] idf here is defined as $idf_i = \frac{|\text{All pred-arg slots}|}{|\text{pred-arg slots filled by i}|}$

Some other classes containing verb arguments with a clear semantic relationship to each other can be found in the following:

Class 1 consists among others of the first arguments of:
think, rethink, believe, know, consider, reconsider, understand, remember, respect, underestimate, value, view, visualize, respect

Class 4 contains objects of verbs related to financial transactions and consumption such as: buy, sell, calculate, acquire, afford, auction, buy_up, buy_out, cut_down, exchange, lose, begin, continue, feed, keep, maintain, market, obtain, regain, retain, trade, use.

Class 4 also contains subjects of verbs such as:
diminish, decrease, descend, crush, double, eat, eat_up, end, extend, fail, discharge, dismiss, dispatch, dissolve, distort, exhaust, launch, multiply, pay, plunge, profit, quadruple, shrink, spend, yield, triple.

2.4 Semantic typing of verb arguments

Clustering verb argument slots as described above leads both to the semantic grouping of verbs as well as the indirect semantic typing of the words that feature as arguments to the verbs. For the latter, details regarding membership of the 32 classes of verb argument positions were combined with information about which words appear in which slots, so that each term was assigned to the corresponding classes. This inevitably resulted in a word belonging to more than one class. For example, the term "spokeswoman" is a filler of verb argument slots in classes 6, 9, 7 and therefore belongs to the homonymous classes. However, its frequency in each class will differ so that combined with the ipf[4], the respective tf-ipfs give a better idea of how meaningful class membership is. For example, the tf-ipfs for the term"spokeswoman" are 0.0075, 0.00275 and 0.0005 for classes 6, 9 and 7 respectively, making class 6 the most representative class for this word.

By looking at the 15 highest ranking terms in each class, where rank is determined by descending tf-ipf, we attempted to give labels to the 32 classes of verb arguments. The labels originated from the 3–4 terms with the highest frequency among the top 15 words and are shown below:

```
class label
0 proposition
1 company_organisation
2 unspecified_someone
3 proposition_truth_profit_patient_impact
4 percentage_money_income_revenue_stock_share_asset
5 percentage_mony_numXpression
6 spokesman_company_person_analyst
7 income_revenue_net_rate_cost_stock
8 place_step_effect_loss_action
9 proposition_company_spokesman_revenue_analyst
10 proposition_stake_rate_percentage
11 proposition_percentage_sure_decision_bid
12 year_percentage_quarter_index
13 reporter_dividend_money_percentage_analyst
14 percentage_proposition_numXpression
```

[4]ipf (inverse predicate frequency) is defined in the same way as idf

```
15 proposition
16 percentage_stake_demand_money_rate_cash_capital
17 proposition_projection_rate
18 proposition_trading_pressure
19 proposition_table_corner_board_tide
20 proposition_percentage_public_private_high_low_numXpression
21 government_civilian_unspecified
22 proposition_unspecified_game_role_cash_company_agreement
23 percentage_proposition_numXpression
24 percentage_proposition_date_profit
25 director_court_partner_company
26 proposition_contract_profit_demand_requirement_proposal
27 demand_problem_leak
28 year_month_time
29 proposition_money_percentage_share_stock
30 year_time
31 fund_proposal_investor
```

As can be seen from the above, obtaining a clean-cut label reflecting the meaning of the contents in each argument class is a non-trivial process; there seems to be a significant amount of sense variance within a class and overlap between classes.

2.5 Adding hierarchy to the semantic typing

In order to obtain a sense of the extent of overlap and similarity between classes, we computed a similarity matrix consisting of the pairwise similarities between each of the 32 classes where similarity between classes was defined in terms of the **overlap coefficient**:

$$sim(A,B) = \frac{|A \cap B|}{min(|A|,|B|)}$$

Where $|A \cap B|$ is the number of words in both A,B and $|A|$,$|B|$ are the number of words in classes A,B respectively.

The overlap coefficient considers classes to be similar when one subsumes or nearly subsumes the other. Hierarchical clustering was performed on the the basis of the overlap similarity matrix, using euclidean distance as the distance metric. The result is the cluster dendrogram below, which illustrates the relation between classes a lot more clearly than a set of flat labels can and allows for a generalisation hierarchy of the senses reflected by the classes' semantic types.

Even though it is difficult to designate human-friendly labels to the classes that represent their meaning in a straightforward manner, we will show that these classes can be used reliably to automatically assign semantic types to the arguments of verbs.

First, we combined the information about class membership of verb argument slots to create patterns of the form:

ARG1 VERB1 (ARG2) (ARG3)

As verb-argument slots were assigned to each class with probability 1 (see Section 2.2) and we made the "one sense per corpus" assumption, there is just one pattern for each verb. Thus, for example, the pattern for the verb 'report' is the following:

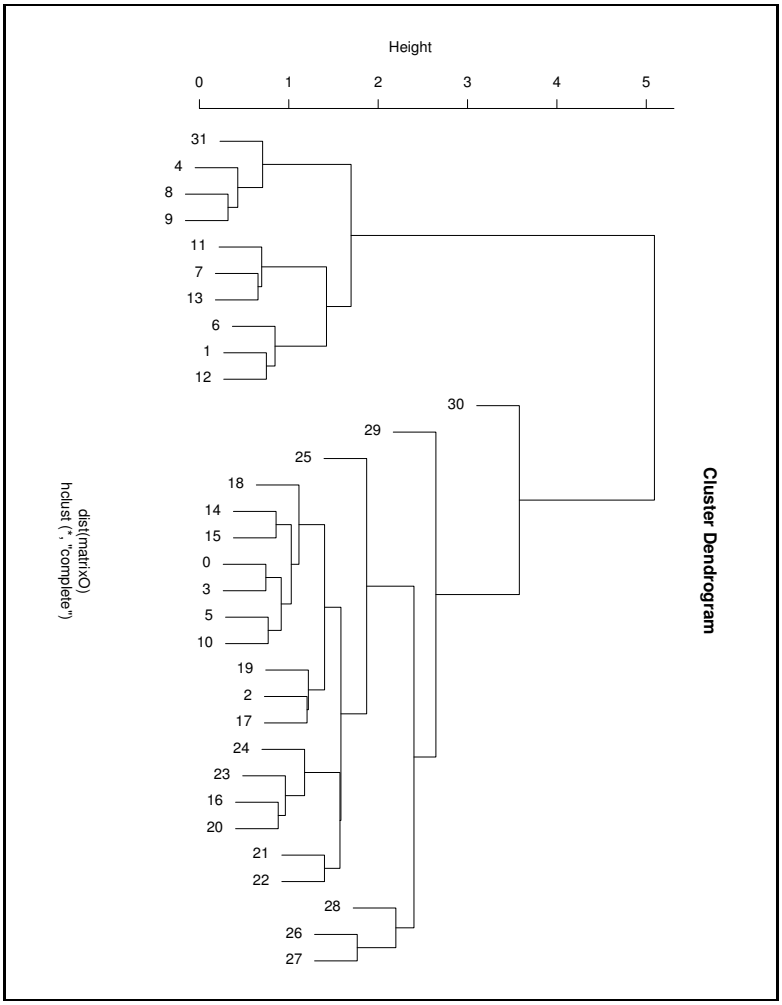

Figure 1: Class dendrogram

[1 report 4 (10)], which takes the following form when replacing class IDs with tentative semantic labels:

[*company_organisation*] report
[*percentage_money_income_revenue_stock_share_asset*]
[*proposition_stake_rate_percentage*]

However, this does not mean that a person cannot be the 1st argument of report; there is overlap between classes 1 and 6 (the major person class) and Figure 1 shows they are closely linked. Such proximity of classes is considered during pattern evaluation (Section 3).

The patterns were stored in a MySQL database. They are partly modelled on the 'Corpus Pattern Analysis' model described in Pustejovsky et al. (2004). These are syntagmatic patterns representing a selection context for the predicate they include, which determines the sense of the latter although CPA Patterns as defined by Pustejovsky et al. (2004) and Rumshisky and Pustejovsky (2006) are in fact rather more detailed than our patterns.

3 Results and Evaluation

To evaluate the semantic types assigned by the automatically derived classes as well as the transferability of the derived CPA-like patterns to unseen instances, we performed a pilot study where we applied the patterns to two randomly selected articles from the on-line versions of the WSJ and the FT from March 2008. We believe this to be a useful test for the validity of the patterns since the new articles are guaranteed to be distinct from the training WSJ data of the 90s, while still belonging to the same domain. We parsed the article using the CCG parser (Clark and Curran, 2007) and concentrated on its RASP option (Briscoe et al., 1997) output, consisting of dependency relations. Since our patterns concern the semantic typing of verb arguments, we focussed on the relations ncsubj (non-clausal subject), dobj (direct obj) and iobj (indirect obj) between a verb and the respective argument position. We ignored erroneous parses[5] as well as copular predicates with the verb to 'be', since the CCG parser's dependency relations did not maintain the connection between 'be' and the adjective or participle, making it clumsy to automatically link arguments in the way we need to.

We then followed the evaluation procedure below, where for each verb-argument pair token in the evaluation set:

1. We looked for a pattern in the database matching the verb-argument relation and augmented the count for recall if a match was found for the right verb.

2. We obtained the type (that is, the class ID) that the pattern assigns to the argument filler word. We then checked the latter in the database, to see which classes it belongs to as well as its freq, tf-idf for each class. Determining which should be the correct, gold standard class of a word given the 32 classes is very difficult considering the class overlap. Therefore, the three highest ranking classes were taken as describing the correct semantic type for the word. Here rank is defined by looking at the 10 first classes where the term has the highest tf-Idf and returning the 3 of these with the highest frequency.

3. If the type assigned to the argument filler matches any of the 3 classes-semantic types, we assumed the type assignment is correct.

4. If the type returned was not among the 3 correct semantic types, we looked at the cluster dendrogram from the previous section and counted the distance between the correct and returned types. If the correct type and the returned type are in the same cluster at the same level, we count the distance as 1. If we need

[5]This can be justified by the fact that we are evaluating the patterns, not the system for producing the dependency relations for evaluation

to go up a level from the returned type for them to be in the same cluster the distance is 2, if two levels, the distance is 3.

5. Proceeded to the next verb-argument pair.

To illustrate the assignment of semantic types through the application of the patterns and the ensuing evaluation procedure, we consider two example verb argument relations from the WSJ text, namely 'dobj shows declines' and 'ncsubj dropped indexes'. In the first case, we looked in the database for a pattern of the verb 'show'. The matching pattern is ' 6 show 4 14', which assigns semantic type 4 to the object of the verb, 'declines'. When looking up the noun 'decline' in the database, the 3 types constituting its correct semantic type are 9, 8, 4. Since 4, the type allocated by the pattern is among them, we consider this to have been the correct assignment of semantic type. For the second example, the pattern available in the database of the verb 'drop' is '9 drop 8 28', which means that the pattern assigns type 9 to the word 'index'. However, when we look up the word 'index' in the database, the correct semantic types for it are 7,12,4. We check in the cluster dendrogram to calculate the closest distance between type 9 and types 7,12,4 which is 2 steps, between classes 9 and 4. The semantic type assignment is therefore considered once more correct.

There were 46 distinct verbs and 78 distinct verb-argument relations that met the criteria for evaluation (out of 119 extracted predicate argument relations) in the WSJ article. For the FT article the corresponding numbers were 25 and 53 respectively (the latter out of 129 predicate-argument relations). The difference in these figures can be due to the size of the articles (6,002 words for the WSJ as opposed to only 2,702 for the FT one) as well as the preference for nominal predicates and nominalisations in the FT article.

A verb pattern existed for each of the verb-argument relations, which gave a perfect recall, 78/78, 53/53 (100%). This is gratifying since the patterns seem to cover adequately the financial domain, given that the test data come from two different newspapers. When allowing a distance of up to 3 between the assigned and correct classes precision was 60/78 (76.9%) for the WSJ and 33/53 (62.2%) for the FT article. For example, in the predicate 'oversees Mac', 'Mac', which is a company, was allocated to class 13 by the patterns whereas the correct class should have been one of 6, 9, 1. The distance between classes 13 and 6 is 3 steps, whereas 'company' features in both classes with tf 0.0008 and 0.011 respectively. The precision was reduced to 55/68 (70.5%) and 30/53 (56.6%) if we only allowed up to 2 steps (e.g. in 'index fell' 'index' was assigned to class 9 where its tf is 0.0014, as opposed to 4 where its tf is 0.0016). Precision fell further to 41/68 (53%) and 26/33 (49%) respectively for up to 1 steps (e.g. where in 'reported at 75' the iobj '75' was classified as being in class 10 (tf 0.0004) as opposed to 5 (tf 0.004). For strictly exact matches, precision was 33/78 (43%) for the WSJ and 21/53 (39.6%) for the FT (e.g. 'director' in 'director said' being assigned to class 6 where the correct type is defined by classes 25,12,6).

The results between the two articles are definitely comparable. However, it is difficult to tell whether the observed difference at the upper end is indeed statistically significant and to what extend the difference between British English and US English plays a role here. Nevertheless, even though the evaluation was only performed on a small scale, we consider the results to be at the very least, encouraging, since the

texts we tested the patterns on were picked at random from the domain of financial news. The perfect recall would suggest that the verb patterns provide reasonably full coverage of the domain, while we can assign informative fine-grained semantic types to arguments with a reasonable degree of precision. Of course, a larger evaluation would be desirable, as would some task-related measure of how much this semantic typing helps in accurate processing. We hope to do this in future work.

4 Related Work

The literature on acquiring semantic classes of words is very extensive. It is mostly motivated by WSD and WSI where the aim is to discover or be able to differentiate between different senses of a target word. Pereira et al. (1993) describes a method for clustering words according to their distributions in particular syntactic contexts. Nouns for instance are classified according to their distribution as direct objects of verbs, where it is assumed that the classification of verbs and nouns co-varies. In our approach we also make this assumption and nouns are clustered indirectly by first grouping together the verb argument slots they fill. Clustering in both cases is probabilistic with the assumptions that members of the same cluster follow similar distributions or in our case a joint distribution.

Phillips and Riloff (2002) and Pantel and Lin (2002) also describe work on clustering nouns to derive semantic classes. Work more directly comparable to ours includes Schulte im Walde (2003, 2006) who presents a method for clustering German verbs by linguistically motivated feature selection. Evaluation against a manually annotated gold standard showed that syntactic subcategorisation features were most informative whereas selectional preferences added noise to the clustering. However, the author concludes that there is no perfect choice of verb features and that some verbs can be distinguished on a coarse feature level while others require fine-grained information. Korhonen et al. (2006) also use syntactically motivated features to cluster together verbs from the biomedical domain and in more recent work (Sun et al., 2008) showed that rich syntactic information about both arguments and adjuncts of verbs constitute the best performing feature set for verb clustering.

Gamallo et al. (2007) follow a similar approach to Pantel and Lin (2002) where an initial set of specific clusters, containing manually chosen terms representative of the domain as well as their lexicosyntactic contexts, are aggregated to form intermediate clusters to which hierarchical clustering is applied for further generalisation. A very interesting aspect of this work is that concept-clusters have a dual nature, consisting both of words-terms (extension) and their lexico-syntactic contexts (intension). As is the case in our approach, cluster formation is twofold, by grouping together words according to the contexts they appear in but also by clustering contexts based on the words they share though this is mentioned as future work in Gamallo et al. (2007). However, in earlier work Gamallo et al. (2005) cluster together similar syntactic positions in Portuguese derived automatically and each cluster represents a semantic condition. Words-fillers of the common position are used to extensionally define the particular condition. Clusters are formed in two stages, where first the similarity between any two positions is calculated in terms of their common word fillers, the 20 most similar ones for each position are aggreggated and the intersection of common words kept as features. Next, basic clusters are agglomerated according to the amount

of shared features. The result is a lexicon of words with syntactico-semantic requirements applied successfully to PP-attachment.

The current work has a different agenda in that it aims to obtain semantic classes of nouns that feature as verb arguments. This information is combined to form selection contexts for verbs, similar to CPA patterns (Pustejovsky et al., 2004), which are then evaluated on the assignment of semantic types. However, whereas our patterns are obtained in a fully automated way, CPA patterns are acquired semi-automatically after the initial manual construction of core verb subcategorisation frames.

5 Summary and Future Work

We have presented a method for automatically acquiring domain-specific selectional restrictions for verbs in terms of semantic typing of their arguments. This was achieved by clustering together verb argument slots sharing the same filler words after obtaining all predicate-argument relations in the corpus. This also resulted in the semantic grouping of nouns, which instantiate the verb arguments. The clustering method used was Autoclass, an extension of the mixture model. We combined the information from the clusters of nouns and verb-argument slots to create contextual verb patterns. The latter were evaluated on a text chosen at random from the same domain and achieved perfect recall and reasonably high precision.

As this pilot study showed that fine-grained domain-specific semantic patterns for verbs can be obtained automatically, we would like to port the approach to a domain where fine-grained typing is of paramount importance. This is the case with the biomedical domain, where for instance verbs of biological interaction, such as inhibit or activate are semantically underspecified (Rumshisky et al., 2006; Korhonen et al., 2006). However, the specific biological interactions come only through the details of the actual arguments participating in the interaction (Rumshisky et al., 2006). We would also like to experiment with different clustering methods and use more sophisticated linguistically motivated filters for feature selection.

Acknowledgements

We would like to thank Rachele de Felice for her assistance and Stephen Clark & Rada Mihalcea for their useful comments. This work was partially funded by the Companions project (`http://www.companions-project.org`) sponsored by the European Commission as part of the Information Society Technologies (IST) programme under EC grant number IST-FP6-034434 and the ART Project (`http://www.aber.ac.uk/compsci/Research/bio/art/`) funded by the Joint Information Systems Committee (JISC).

References

Baker, C. F., C. J. Fillmore, and J. B. Lowe (1998). The Berkeley Framenet project. In *Proceedings of the COLING-ACL*, Montreal, Canada.

Bos, J., S. Clark, M. Steedman, J. Curran, and J. Hockenmaier (2004). Wide-Coverage Semantic Representations from a CCG parser. In *Proceedings of*

the 20th International Conference on Computational Linguistics (COLING-04), Geneva,Switzerland, pp. 1240–1246.

Briscoe, E. and J. Carroll (1997). Automatic extraction of subcategorisation from corpora. In *Proceedings of ACL ANLP 97*, pp. 356–363.

Briscoe, E., J. Carroll, and R. Watson (1997). The Second Release of the RASP System. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, Sydney, Australia.

Cahill, A., M. McCarthy, J. van Genabith, and A. Way (2003). Quasi-Logical Forms for the Penn Treebank. In I. v. d. S. Harry Bunt and R. Morante (Eds.), *Proceedings of the Fifth International Workshop on Computational Semantics, IWCS-05*, Tilburg, The Netherlands, pp. 55–71.

Cheeseman, P. and J. Stutz (1995). Bayesian classification (Autoclass): Theory and results. In P. S. U. Fayyad, G. Piatesky-Shapiro and R. Uthurusamy (Eds.), *Advances in Knowledge Discovery and Data Mining*, pp. 153–180. Menlo Park, CA: AAAI Press.

Clark, S. and J. Curran (2007). Wide-Coverage Efficient Statistical Parsing with CCG and Log-Linear Models. *Computational Linguistics 33(4)*, 493–552.

Clark, S. and D. Weir (2002). Class-Based Probability Estimation Using a Semantic Hierarchy. *Computational Linguistics 28(2)*, 145–186.

Gamallo, P., A. Agustini, and G. Lopes (2005). Clustering Syntactic Positions with Similar Semantic Requirements. *Computational Linguistics 31(1)*, 107–146.

Gamallo, P., G. Lopes, and A. Agustini (2007). Inducing Classes of Terms from Text. In *Proceedings of TSD 2007*, pp. 31–38.

Hanks, P. and J. Pustejovsky (2004). Common Sense About Word Meaning: Sense in Context. In *TSD 2004*, pp. 15–18.

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

Korhonen, A., Y. Krymolowski, and N. Collier (2006). Automatic Classification of Verbs in Biomedical Texts. In *Proceedings of AC-COLING 2006*, Sydney, Australia.

Korhonen, A. and J. Preiss (2003). Improving Subcategorization Acquisition using Word Sense Disambiguation. In *Proceedings of ACL 2003*, Sapporo, Japan, pp. 48–55.

Liakata, M. and S. Pulman (2002). From Trees to Predicate-Argument Structures. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei, Taiwan, pp. 563–569.

Liakata, M. and S. Pulman (2004). Learning Theories from Text. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, Geneva, Switzerland.

Miller, G. A. (1995). "Wordnet: a lexical database for English.". In *Communications of the ACM*, Volume 38 (11), pp. 39 –41.

Pantel, P. and D. Lin (2002). Concept Discovery from Text. In *In Proceedings of COLING 2002*, Taipei, Taiwan.

Pereira, F., N. Tishby, and L. Lee (1993). "Distributional clustering of English words.". In *Proceedings of the 31th Annual Meeting of the Association of Computational Linguistics(ACL 93')*, Columbus, Ohio.

Phillips, W. and E. Riloff (2002). Exploiting Strong Syntactic Heuristics and Co-Training to Learn Semantic Lexicons. In *Proceedings of EMNLP 2002*.

Pustejovsky, J., P. Hanks, and A. Rumshisky (2004). Automated Induction of Sense in Context. In *COLING 2004 5th International Workshop on Linguistically Interpreted Corpora*, Geneva, Switzerland, pp. 55–58.

Rumshisky, A., P. Hanks, C. Havasi, and J. Pustejovsky (2006). Constructing a Corpus-based Ontology using Model Bias. In *FLAIRS 2006*, Melbourne Beach, Florida.

Rumshisky, A. and J. Pustejovsky (2006). Inducing Sense-Discriminating Context Patterns from Sense-Tagged Corpora. In *LREC 2006*, Genoa, Italy.

Schulte im Walde, S. (2003). Experiments on the Choice of Features for Learning Verb Classes. In *Proceedings of EACL 2003*, Budapest, Ungarn.

Schulte im Walde, S. (2006). Experiments on the Automatic Induction of German Semantic Verb Classes. *Computational Linguistics 32(2)*, 159–194.

Sun, L., A. Korhonen, and Y. Krymolowski (2008). Verb Class Discovery from Rich Syntactic Data. In *Proceedings of the 9th International Conference on Intelligent Text Processing and Computational Linguistics*, Haifa, Israel.