Cluster-Based Pattern Recognition in Natural Language Text

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science

by

Shmuel Brody

under the supervision of Prof. Naftali Tishby

August 2005

Acknowledgements

I would like to thank my adviser Prof. Tishby for his guidance and assistance in producing this work, and for his suggestions and positive input. I would also like to thank Beata Beigman Klebanov for her constant help and advice throughout this work, including (but definitely not limited to) the contribution of the parsed data used here. Also deserving of thanks are my family, for their support, and especially my grandmother, Rose Brody, for her confidence in my achievements.

Abstract

This work presents the *Clustered Clause* structure, which uses information-based clustering and dependencies between sentence components to provide a simplified and generalized model of a grammatical clause. We show that this representation, which is based on dependencies within the sentence, enables us to detect complex textual relations at a higher level of context. The relations we detect are of interest in themselves, as linguistic phenomena, and are also highly suited for use in certain linguistic and cognitive tasks. We define and search for several types of patterns, moving from basic patterns to more complex ones, from patterns within the sentence to those involving entire sentences. Examples of recognized patterns of each type are presented, and also descriptions of several interesting phenomena detected by our method. We assess the quality of the results, and demonstrate the importance of the clustering and dependency model we chose. The principles behind our method are largely domain-independent, and can therefore be applied to other forms of structured sequential data as well.

Table of Contents

1 Introduction

1.1 The Problem

This work is concerned with the problem of detecting patterns in sequential data. When we deal with sequences where each point in the sequence can have one of a very large number of values, such patterns are often difficult to detect. The difficulty stems mainly from the problem of data sparseness, meaning that our sequence is not (and usually, cannot feasibly be) long enough to give a true representation of the value distribution. The large number of values presents another problem: since we are usually looking for patterns which should be applicable to a large part of the data, finding a pattern which applies to a small number of values is of little use to us. The procedure we present here is designed to solve both these problems, and facilitate the pattern detection task. It combines the use of clustering via mutual information with a model chosen to fit both our specific pattern detection task and the data we deal with.

An important example of such a situation is pattern detection in text. If we view the text as a sequence of sentences, it is easy to see that finding patterns between sentences is very difficult. We hardly ever encounter the same exact sentence more than once, so at first glance, no patterns exist between whole sentences. We are well aware, however, that patterns *do* exist, but on the semantic level, rather than the purely lexical one. Since semantic information is not usually available, we need to derive the underlying semantics of the text from the text itself. Our method attempts to do exactly that. It extracts a semantic representation of key information in the text, and uses that representation to detect high level relationships.

We aim to show the success of our procedure in handling the detection task, and the contribution of the clustering and modeling to that success. Our results are presented with these aims in mind. We also show and discuss pattern-related phenomena which were detected in the data with the aid of our approach. Though we chose to apply our procedure on natural language data, the principles behind the method are applicable to other data as well.

1.2 Overview

The rest of the paper is organized as follows: this chapter presents an overview of previous work in the field. It explains the relevance of this paper within that context, and illustrates some of the possible uses of our method. Chapter 2 presents a general description of the methods and tools we used in our procedure. It also describes the clause model we use, the types of patterns we search for, and our method for evaluating the results. Chapter 3 describes in more detail how we used the methods described in the previous chapter, and applied them to our task. Chapter 4 presents the results of our procedure and points out some interesting phenomena detected by our method. Chapter 5 concludes with a discussion of the procedure and its implications and ideas for future improvements and modifications.

1.3 Related Previous Work

1.3.1 Syntax and Distributional Information as Measures of Semantics

The relationship between textual features and semantics and the use of syntax as an indicator of semantics has been widespread. Following the idea proposed in Harris' *Distributional Hypothesis* [1], that words occurring in similar contexts are semantically similar; many works have used different definitions of *context* to identify various types of semantic similarity. Levine, in her work on the classification of English verbs [2], uses the concept of *alternations* as a tool in the study of a verb's meaning and its syntactic behavior. An *alternation* is a relation between a pair of similar syntactic frames, involving a change in the number or order of the arguments the verb accepts. For example, the *causative-inchoative* alternation: in verbs that undergo this alternation the subject of the intransitive verb is related to the object of the transitive, as in the two sentences "A boy broke the window", "The window broke". Levin views the types of syntactic alternations which a verb can take as indicative of its meaning, and uses this feature set to divides verbs into classes sharing a common core of meaning. Zickus [3] compares Levin's classification to that of WordNet, which is manually constructed and serves as a common reference point for semantic similarity (see below). She finds the there is consistency in the association of semantic components with particular types of alternations, and concludes that Levine's alteration methodology is an appropriate one for testing semantic distinctions. Douglas *et al.* [4] use Levine's system for verb classification in other languages, demonstrating that the connection between syntax and semantics is global, and not limited to the English language structure.

Hindle [5] uses a mutual-information based metric derived from the distribution of subject, verb and object in a large corpus to classify nouns, and claims this metric reflects semantic relatedness. Periera, Tishby and Lee [6] cluster nouns according to their distribution as direct objects of verbs, using information-theoretic tools (the predecessors of the Information Bottleneck tools we use in this work). They suggest that information theoretic measures such as relative entropy, or predictive power with regard to a target variable distribution, can also measure semantic relatedness.

Grefenstette [7] compared two types of automatically constructed thesauri to manually constructed ones. The first construction method used sophisticated language tools to extract the syntactic context of each word throughout the corpus. The second relied on simple proximity information. In the first method, the corpus was divided into lexical units via a regular grammar; each lexical unit was assigned a list of context-free syntactic categories, and a normalized form. Then a time-linear stochastic grammar selected a most probable category for each word. A syntactic analyzer algorithm chunked nouns and verb phrases and created relations within chunks and between chunks. A noun's context became all the other adjectives, nouns, and verbs that enter into syntactic relations with it. The second method used much simpler window-based statistics to group words together. A noun's context was defined as all the words which could be nouns, adjectives or verbs in a fixed-size window around it.

The comparison showed that the first technique performed significantly better than the second one, providing more exact information on the few hundred most frequent nouns. This shows that syntactic information plays an important part in discovering semantic similarity. On the other hand, the second technique performed better in the case of rare words, since is allowed the extraction of more information than was available in the immediate grammatical context, even though this information was much less exact. In other words, the overall context of the paragraph or document (which was captured better in the second method, especially with large windows) was helpful in the case of rare words, where there were not enough instances from which to draw exact semantic conclusions.

The theory of *Latent Semantic Analysis* (LSA) [8,9] proposes a very strong connection between distributional information and semantics. In LSA, semantic relatedness is measured using a high-dimensional semantic space. This space is created by performing *Singular-Value Decomposition* (SVD) on a matrix constructed from analyzing a large example corpus for various types of co-occurrence. Terms and larger textual units, such as sentences, can be represented as vectors in this space, and can be compared using vector distance measures. The theory behind LSA claims that similarity in this representational space indicates semantic similarity. The theory suggests that high-level knowledge can be inferred from relatively simple distributional information.

The types and use of syntactic relations can be roughly divided into two groups: relations defined in terms of templates or patterns, and relations defined by similarity measures based on some kind of syntactic or distributional feature set. These two perspectives are described in the next two sections.

1.3.2 Relations from Patterns and Templates

One of the ways to discover relationship between words is to use fixed templates. Words are considered to be related if they are connected through a fixed template in a given body of text. Templates can also be used to classify words, by grouping together words that fill a specific role in a fixed template. The most straightforward examples are templates such as "X is a Y" appearing in the text, which clearly demonstrate a relation of hypernymy or hyponymy between X and Y. Another example might be using the template "X of Y" to group together all the words which fill the part of X for a given Y (for instance, "wheels of a car", "doors of a car" and "roof of a car").

There have been several works done to automatically detect specific relation types using appropriate templates. Defining the appropriate templates which accurately reflect the relationship being sought is part of the problem, and several approaches have been proposed. The "Is-A" (hypernymy/hyponymy) relation is probably the most popular. Hearst [10] used seven predefined lexico-syntactic patterns which capture this relation. She starts off with three simple patterns observed in the text, and uses the matching word arguments to (manually) search for other templates, which are generalized as much as possible by hand. Iwanska *et al.* [11] use two very simple templates for this task: "such as", and "like". The novelty in their work lies in their method for resolving the template components and boundaries very efficiently, without use of complicated syntactic analysis.

Berland and Charniak [12] used templates to discover the '*part-of-a-whole*' (meronym) relationship. First, an example list of parts and their whole was manually generated, and then the corpus was searched for these examples, looking for the part and its whole in close proximity. When these were found, the patterns linking the part and the whole were added to a list of patterns, and used for further search. The use of a seed list of related words replaces the group of seed patterns used by Hearst, so that the process becomes automatic at an earlier stage.

Velardi *et al.* [13] use patterns to detect causal relations. Their method is similar to Berland and Charniak's, involving the production of a seed list of nouns which are cause-and-effect and subsequent detection of patterns linking these pairs from instances in the text. The seed list was constructed with the help of WordNet, using the "*Cause-To*" relation defined between WordNet's synonym sets. The templates sought were simply "NP1 VP NP2", and several validation methods were required to ensure that the cause relationship was actually present.

The template-based method of detecting relations has several important advantages. Templates are much easier to detect in large bodies of text, and therefore the algorithms which employ them are much quicker (as shown by Pantel *et al.* [14]) and computationally simpler than their similarity-based counter parts. Very little preprocessing is needed, and part-of-speech tagging is often enough. Also, if the templates are specific enough, their precision is very high. On the other hand, template-based methods often suffer from poor recall, as a result of data sparseness. Specific patterns are relatively rare, so many existing relations may be missed. One of the ways to overcome this problem is to use the results of template-based Google queries as corpora (Markert *et al.* [15], Agirre *et al.* [16], Cimiano *et al.* [17]).

1.3.3 Feature Sets and Similarity Measures

Another method of using syntax to derive semantic relationships is to define a set of syntactic or distributional features, and use this feature set for some sort of similarity/relatedness calculation. Many different feature sets have been proposed, usually with regard to some notion of the word's context and suited to a specific task. So, for instance, we have feature sets which are defined by the occurrence of the word in a set of documents, as in the work of Akrivas *et al.* [18], where the similarity between two index words was defined by the number of times they co-occurred in the same document. Similarly, Mandala *et al.* [19] use co-occurrence of words in the same document to determine the similarity, but here the distance measure was the mutual information between the words, calculated from these co-occurrence probabilities.

Co-occurrence data between the words themselves is also widely used as a feature set. In some cases, the co-occurrence event is simply when two words occur within a specified distance of each other, with no consideration of language structure. Other feature sets make use of more sophisticated syntactic relations, such as Lin's [20] *modifier-modified* feature set, where a word's feature vector is made up of the number of times it modified or was modified by another word, also taking into account the type of modification relation. The dimension of this feature set is enormous (containing millions of possible features), which presents a problem for many traditional clustering methods. Pantel and Lin [21] propose the *Clustering-By-Committee (CBC)* algorithm partially in order to solve this problem. This algorithm first finds small groups of very similar objects, well dispersed in the feature space, and assigns them as the committees in charge of defining the cluster centroids (a centroid is calculated by averaging the feature sets of the committee members). Once the centroids have been defined, all other objects are assigned to the clusters based on their similarity to the cluster centroids. This avoids the need to compute the similarity between every pair of objects, replacing it with comparisons between each object and a small number of centroids.

Another solution to the high-dimensionality problem is to use subsets of this feature set. In fact, direct grammatical relations such as subject-verb and verb-object, or relations between parts-of-speech, such as the relation between a noun and its adjectives, can be seen as subsets of this feature set, and have been very widely used, as in the work of Hindle and Periera *et al.* mentioned above, and by Mandala *et al.* in a different part of the paper mentioned previously. This solution also deals with another problem of Lin's complete feature set – the fact that all the relation types are considered equal, even though some are clearly much less informative than others.

Once a feature set is defined, there are three main approaches regarding its use for calculating similarity or relationships. The first of these regards the feature set as a vector in a high dimensional space, and uses standard mathematical distance functions to calculate similarity. Such a measure was suggested by Patwardhan [22] to measure similarity between WordNet synonym sets, and is used by Foltz *et al.* [23] to measure similarity between sentences using a high dimensional space based on LSA (see above). The second approach uses information-theoretic measures, such as mutual information based on co-occurrence vectors, or distance measures between distributions (such as *Jensen-Shannon divergence* or the *Kulbeck-Leiber* distance measure). This approach was employed in the works of Hindle, Periera *et al.* and Mandala *et al.*, as mentioned above. The final method takes a set-theoretic approach, converting each object's feature vector into a matching set, and looking at the inclusion relations between these sets. Cimiano *et al.* [24] take this approach, and use Formal Concept Analysis (FCA) (Ganter *et al.* [25]) and its accompanying definition of the *superconcept-subconcept* relationship based on the feature sets. Another example of the set-theoretic approach is the work of Sanderson and Croft [26], who assign to each term a set of all documents in which that term appeared, and create a hierarchal structure where term A is considered more specific than term B if A's set of documents is a subset of B's.

Some similarity measures are based directly on some humanly annotated list of semantic relations. Such is the case with the similarity measures proposed by Resnik [27], by Leacock and Chodorow [28], and by Hirst and St. Onge [29], which use various features of the WordNet semantic network to determine similarity. These measures circumvent the need to derive the relations from the corpus, but this is also their disadvantage, since they use global relationships, which are disconnected from the context of the corpus they are used on. Some corpus-specific relationships may be missed, and it is quite likely that some of the words in the corpus will not be contained in the semantic network at all. The strongest drawback is, of course, the availability of a manually constructed database of sufficient precision and completeness.

1.3.4 Uses of Similarity and Relatedness Measures

Measuring similarity or relationships between words, or grouping words into classes or categories, is one of the most basic prerequisites of most natural language tasks (and of many cognitive tasks as well). For instance, similarity measures are used by McCarthy *et al.* [30] for one of the most basic NLP tasks - word sense disambiguation. Shutze [31] makes a fine distinction between sense *disambiguation*, where the task is to choose the correct sense from a given list of senses, and sense *discrimination*, where we want to know if two appearances of a word have the same sense. He proposes a method using distributional statistics to accomplish the latter, overcoming the need for manually annotated word sense lists. Lin [20] describes the closely related task of inferring the meaning of an unknown word from its context as the main use of his word similarity measure. Weimer-Hastings *et al.* [32] use both semantic and syntactic features to infer the meaning of unknown verbs. Poesio *et al.* [33] used corpus-based tools for detecting relationship between words (they focus on the *part-of-a-whol*e relation), for the purpose of anaphora resolution.

Similarity measures are useful in the field of information retrieval, as well. Classifying is used by Akrivas *et al.* (see above) in query expansion, to find documents similar to the direct results of the query, and display them. Bekkerman *et al.* [34] show that word clustering can also improve existing document classifications techniques. In addition, word similarity can be used to expand the queries themselves, by adding words to the query which are similar to the original ones, as in Harris [35].

1.3.5 Semantic Databases

While many works focus on a specific task, and therefore focus more strongly on the specific relation or type of similarity that they consider suitable to that task, other works aim at creating relatedness databases, in which useful forms of semantic knowledge can be stored in machine readable format for easy access. These projects try to address a general need, and are usually concerned with the most basic types of relations, which should be of use in many NLP tasks.

A prominent example of such a knowledge database is WordNet [36], an online lexical reference system. WordNet defines synonym sets, or *synsets*, which are composed of words with a single underlying lexical concept. Each synset has an accompanying gloss – a short dictionary-like explanation of the concept. WordNet provides links of several types between synsets, representing different kinds of semantic relations. The synset nodes and relationship links create a graph structure. WordNet is manually constructed by experts, and its design is inspired by current psycholinguistic theories of human lexical memory.

The VerbNet project [37] provides manual verb classification compatible with WordNet but with explicitly stated syntactic and semantic information. VerbNet uses Levin's verb classes (see above) in the constructions of the lexical entries, but extends and refines them. The VerbNet project relies completely on Levin's formal class definitions, and does not attempt to construct classes automatically from syntactic data.

FrameNet [38] is another manually constructed database which focuses on documenting *lexical units*, which are word-meaning pairs. FrameNet is based on *Frame Semantics* [39], where the basic unit is the *semantic frame*. A semantic frame describes a particular type of situation, object, or event and the entities and objects involved in it, which are called *frame elements*. Words which *evoke* the semantic frame (usually verbs) are lexical units in that semantic frame. FrameNet contains documentation of the range of semantic and syntactic valences of each word in each of its senses, and also links between semantic frames. The project is committed to corpus evidence, and uses computer-assisted annotation of example sentences in the documenting the entries. PropBank [40] is a project with a similar aim, providing semantic frame annotation to the PennTree bank [41]. While FrameNet focuses on the precise description of the frames and the connections between them, providing a more accurate semantic knowledge base, PropBank emphasizes coverage and annotation of the text, and is more suitable as a good training sample of annotated text for learning semantic roles.

In addition to manual construction of such databases, some papers address the task of automatically extending such databases using syntax-based methods, as is the work of Mandala *et al.* which was mentioned above. Other projects aim at fully automatic construction of semantic relation databases based on syntactic information. This was the goal Cimiano *et al*. and of Sanderson and Croft (see above), who created a concept hierarchy from syntactic features. Another example is the VerbOcean project [42], in which Chklovski and Pantel construct a repository of verbs and detect several types of semantic relations between them with the aid of a semi-automatic method that uses lexico-syntactic patterns in Google queries.

There are obvious advantages to manually constructed databases. First, the notion of *semantics* and *meaning* are purely human ones and, as such, can probably be accurately described only by humans. In addition, the annotation is supported by the annotator's (usually expert) knowledge, formal linguistic theories or, at the very least, common sense. Manually constructed databases are used as the standard against which the results of automatic and syntactic methods are measured. Unfortunately, manual construction of such databases is extremely labor intensive and time consuming. Also, the coverage of such databases is far from complete, and they are of limited use in domain-specific applications. Automatically constructed databases, on the other hand, require much less effort to construct and maintain, and can usually be adapted to specific domains with ease. The coverage of the database depends on the corpus from which it is constructed, which is usually similar to the one on which the information in the database will be used. The biggest drawback of the automatic approach is the problem of precision – there is no guarantee that the results of the method are correct and accurate. Both the manual and the automatic techniques suffer from the problem of limited recall – there is no way of knowing that all the objects, concepts or relations of interest have been captured.

1.3.6 Relationships Involving a Higher Level of Context

Since in this paper we deal with relations at the clause level, rather than those between individual words within the clause, this section will discuss works which deal with higher-level relations. When we move up from the word level, the basic units which are used become less standardized, and depend on the task or on the theories on which the methods are based. We have already mentioned some works which deal with clustering and similarity between whole documents. For this task, the *bag-of-words* approach has been shown to be successful in classifying documents [43], and some improvement has been shown with the use of clustering (Bekkerman *et al.,* see above). These kinds of relations are of less interest to the work presented here, which deals with syntactically inferred relations involving nearby sentences or clauses, an intermediate level between individual words and whole documents.

At a slightly lower level, M. Hearst [44] presents two algorithms that use lexical cohesion relations to divide texts into multi-paragraph segments in a way that reflects the sub-topic structure of the document. The algorithms differ in their methods for determining lexical cohesion. The first compares blocks of text using a fixed size window, and determines how similar they are lexically. The second uses active chains of repeated terms to determine where segments begin and end. Both these methods use words as feature sets, similar to the *bag-of-words* approach for documents. Stronger syntactic relations are not used, and may not be needed for the task of sub-topic segmentation. While this work deals with relationships within the document, relationships between adjacent paragraphs are still at a slightly higher level than the one we address in our work.

Chklovski and Pantel, in the VerbOcean project (see above) deal with relations between verbs. They present an automatically acquired network of such relations, similar to the WordNet framework, though the relations are more fine-grained, and the coverage is much wider. Though the patterns used to acquire the relations are usually parts of a single sentence, the relationships themselves can also be used to describe connections between different sentences, especially the *enablement* and *happens-before* relations. Since verbs are the central part of a clause, this work can be viewed as detecting relations between clauses as whole units as well as those between individual words. Web queries were used to overcome the problem of data sparseness. VerbOcean does not (yet) attempt to create generalized relations, or to group verbs into clusters or synonym sets, though this could presumably be done using the *similarity* and *strength* relations which are defined and detected by the system. This approach comes close to the type of clause relations we detect in our work, but we do not limit ourselves to verbs alone, and also attempt to find relations between complete clauses. VerbOcean is still a word-word relationship database, though it addresses longer-context relations through dealing with verbs. In order to better capture such relations, it would probably need to be expanded to deal with entire verb frames (verbs and their arguments).

The DIRT system [45], created by Lin and Pantel, deals with inference rules, and employs the notion of *paths* between two nouns in a sentence's parse tree. The system extracts such path structures from text, and provides a similarity measure between two such paths by comparing the words which fill the same slots in the two paths, taking into account the word distribution in the text (infrequent words are more informative). After extracting the paths, the system finds groups of similar paths. This approach bears several similarities to the ideas described in this paper, since our clause structure can be seen as a specific path in the parse tree (probably the most important one). In our setup, similar clauses are clustered together in the same *Clustered-Clause*, which would be comparable to clustering DIRT's paths using its similarity measure. On the other hand, there are several important differences between our methods and those used in the DIRT system. Our method used only the relationships inside the path or clause in the clustering procedure, so the similarity is based on the clause structure itself. Also, Lin and Pantel did not create path clusters or generalized paths, so that while their method allowed them to compare phrases for similarity, there was no convenient way to identify high level contextual relationships between two nearby sentences. This is one of the significant advantages of clustering over similarity measures, in that it allows a group of similar objects to be represented by a single unit.

Foltz *et al.* [23] deal with textual coherence using *Latent Semantic Analysis* (see section 1.3.1). In this work, the units compared for similarity were whole sentences, and the problem of different representation of the same information is addressed by the use of the high dimensional space, which effectively clusters together terms of similar meaning. This coherence measure was shown to be highly correlated with the comprehension level of human readers. This type of relatedness between sentences is very similar to the one addressed in our work, though the general approach is very different. The matrix used by LSA is based on word-context co-occurrence, and does not employ structural data, or even make use of the ordering of the words. All the different levels of context (sentence, paragraph and document) are used together in the production of the vector space. Our approach, on the other hand, uses relations within the clause context to cluster words, and these clusters are used in turn to detect relations at higher context levels. The higher level similarity thus depends upon that of the lower level. Our approach also avoids an important disadvantage of LSA and other methods which rely on complex mathematical analysis: while these methods often display very good results, the internal representation of knowledge is usually not comprehensible to humans, and it is very hard to 'get a feel' of what is actually contained in the representation.

1.3.7 Patterns Containing Cluster Units

Besides dealing with relations at a higher context level, our work also uses clustering to help overcome data sparseness and to define generalized patterns. While the former use is quite widespread, the latter one is much less common. Pantel and Ravichandran's method [46] for obtaining labels (category names) for clusters can be seen as a form of cluster-based generalization. In this work, following Hearst, several manually defined linguistic patterns implying the '*is-a*' relation between cluster components and another word are used to decide that that word should be a label for the cluster. Obtaining cluster labels is a form of generalization, especially if the label is then used to represent the cluster members in some way.

The work of Riccardi and Bangalore [47] on phrase grammars uses clusters to find common phrases in the text, such as "United States of America", "A T & T", "by and large", etc. In their system, the clustering and the phrase detection are integrated in an iterative process. The clustering uses proximity relations rather than sentence structure, and the patterns they seek are of a specific nature – common phrases.

Maedche and Stabb [48] consider the task of finding general association rules between nouns, in a data-mining setup. The relations they search for are those between individual nouns at the sentence level. They use of the method described by Srikant and Agrawal [49] which travels up a predefined concept hierarchy to generalize the association rules to the highest degree possible while still maintaining the required support and confidence values. Our cluster-based generalization method can be seen as a very limited version of Srikant and Agrawal's, where we are in effect using a two level hierarchy – words and clusters. Using a hierarchal clustering method could provide us with more precise generalization, and is suggested as a possible improvement in the final chapter of this paper.

1.3.8 Novel Aspects of this Work

We have already mentioned that dealing with relationships between whole clauses is a relatively novel idea. The use of clustering as an aid to generalization, though not entirely original, has not been widely used in the field (except maybe in information retrieval, for query expansion). Combining these two ideas is especially pertinent effective in our case, since the generalization also helps us overcome the data sparseness which becomes more acute as the type of relations we seek become more complex.

Another novel feature in our work is the use of structure with clustering. Despite the fact that many of the works mentioned above use some sort of structural information in the data when building feature sets, the structure itself is then usually ignored, and the feature set is treated simply as a regular numeric vector when compared for similarity. There is no attempt to take the structure of the data into account at a later stage. In this work, on the other hand, the structure model is used not only for defining the clustering task, but is also strongly linked to the pattern definition and consequent search.

1.4 Importance and Motivation

As mentioned above, some measure of similarity or relationship between language components is a prerequisite of many basic NLP tasks. Many of the uses mentioned in section 1.3.4 would benefit from a measure of relatedness between higher context units in addition to the usual word-similarity measures. Also, there are some linguistic and cognitive tasks in which a higher-level relatedness measure is essential, or can make a significant difference. We mention some of these tasks in this section.

1.4.1 Cognition & World Knowledge Acquisition

It is widely agreed that pattern recognition plays an important part in human cognition. Humans detect patterns that appear in many types of data, recognize instances of these patterns, and draw the relevant conclusions. One of the most important parts of this task is focusing on the relevant parts of the data, and discarding irrelevant 'noise'. Building a model which is both simple and contains the important data is a big step towards recognizing and using patterns. Another important aspect of the task is reaching a generalized conclusion from a few specific instances. Our procedure indicates a possible solution to the problem of implementing these human cognitive qualities in a computerized setup. Human cognition does, of course, deal with basic similarity between objects, but it also detects patterns and similarity at a much higher level than can be described by single words. The type of patterns we present here are a step in the direction of more complex relations, such as those between simple actions or events.

1.4.2 Automated Rule Acquisition

In the field of Artificial Intelligence, there is a long history of rule based systems for theorem proving (most notably Prolog [50], 1972) or plan construction (mostly extensions of STRIPS [51], 1971). These systems use a set of predefined rules to devise a plan composed of a series of actions that, when implemented, should achieve a given goal. One of the big problems with such systems is the need to specify the rules in advance. In many cases this is done by hand, either by the system designer, using his own world knowledge, or with the help of an expert in the field, whose verbal rules are translated by a programmer (to the best of his understanding) into the required rule format [52]. In either case, the task of defining the rules is labor intensive, and is often inaccurate, either because the expert's advice was not understood or translated correctly, or because some of the underlying rules were overlooked by the expert or the programmer, even though they exist 'in real life' and in the expert's knowledge, in some subconscious level.

The *'Subject-Verb-Object'* structure we use in this work is very similar to predicate logic rules used by planning systems, for example *love(John, Mary), has(driver, car)* etc. Instances of these structures in a text and the relations between them can be used as the foundation for a set of rules for such a system. Although many relevant rules may not be specified explicitly in the text, it is also likely that some of the rules which are important to the system but were overlooked by the expert or the programmer will be retrieved by our procedure.

In other cases, where the system is not as goal oriented as the ones described above, we may be interested in introducing general world knowledge to a system. For instance, it may be useful for an automatic travel agent in charge of booking hotel reservations to 'know' that the information "A storm struck Jerusalem today" may be important to a customer's decision when making travel arrangements. Our approach allows the automatic acquisition of some such patterns, and recognition of their instances.

1.4.3 Query Enhancement

With the spread of the World Wide Web, and the enormous amount of data available, data retrieval systems have become very important. Despite their importance, many of the systems are still only word based, and complex queries are treated as simple groups of words. Adding structure to the queries may improve the quality of the retrieval (e.g. requests for 'fire employees' will not return documents containing the sentence such as "The fire in the factory killed two employees").

Another advantage to using a model such as the one described here comes from the clustering effect. Sentences such as "The director informed the worker" and "The manager told the employee" will both be assigned to the same *Clustered-Clause* (see section 2.4) since the subjects, verbs and objects of both sentences belong to the same respective clusters.

This effect can also be used in query expansion – adding elements to a user's query in order to increase the chance of finding relevant information. When given a query, the system may decide to enhance it with similar words or clauses which have a larger likelihood of appearing.

1.4.4 Implication & Entailment

A related issue is the task of recognizing implication, or entailment. In this task, we would like to recognize statements in the text which may imply the truth of a query statement (or hypothesis). For instance, if the system is asked *"Is Bill Gates rich?"* we would like it to know that the sentence *"Bill Gates owns multi-million dollar corporations"* is relevant to the question, and present it to the user. What we are actually demanding is that the system recognize that "*X* owns (expensive) corporations" implies that "*X* is rich". Detecting this type of relation was recognized as a common task of importance in Natural Language Processing, Information Retrieval and Machine Learning, covering a broad range of semantic-oriented inferences needed for practical applications. This task was proposed by Dagan *et al.* as the PASCAL Challenge this year [53].

The system proposed in this paper could be useful in detecting such entailment relations. The chances of finding the required relation expressed explicitly, even in a large body of text, are quite slim, but they increase dramatically when clustered clauses are used, so that many examples containing the same overall meaning are also taken into account. For instance, in the example above, clauses containing "has" instead of "owns" or "wealthy" instead of "rich" are also considered, and all these examples define a generalized relationship pattern.

1.4.5 Anaphora Resolution

The types of patterns we define, especially the third type (see section 2.5) may be well suited for anaphora resolution. We detect anchored patterns between clauses, and retrieve those with statistical significance. Once we have a list of such patterns, they can be used for the task of anaphora resolution. When encountering a clause where the subject or the object are pronouns, we can check the pattern lists to see if it matches one of the patterns with the anchor position being filled by the pronoun. If so, we can then look for the second half of the matching anchored pattern in the vicinity of the first half, and resolve the anaphora by selecting the anchoring noun from the second clause.

Since these patterns are quite complex and statistically significant, the probability that the clause containing the anaphora accidentally matches a pattern is quite small. Consequently, we would expect a high degree of accuracy using this method. In effect, we are using all the other parts of the pattern to predict the correct noun being referenced. On the other hand, it is likely that this method will suffer from low recall due to the rarity of pattern instances.

2 The Work Setup

2.1 The Clause Model

2.1.1 MINIPAR's Sentence Structure

In this work we use Dekang Lin's parser MINIPAR, a descendent of PRINCIPAR [54]. This parser has the advantages of being completely unsupervised, and very quick. This allows us to parse very large amounts of text in a (relatively) short time.

MINIPAR models the sentence structure in a tree-like form. Each word in the sentence modifies exactly one other word, and is modified by one or more words. The modifying relations can be of several, predefined, types. For instance, in the sentence "The boy threw a ball", "ball" modifies "threw" via the "*object-of-verb*" relationship. Each sentence has an abstract root node, which is modified by the verb via the "*i*" relationship. This can be represented as a tree, where the (labeled) edges are the modifying relation, and the words are the nodes.

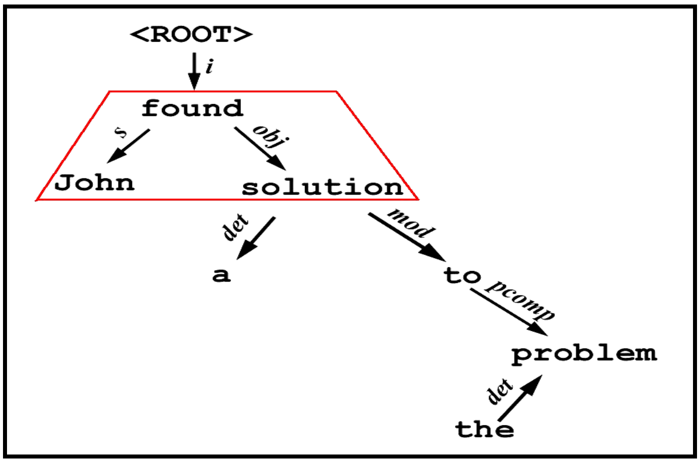

Figure 1 The parsed tree for the sentence "John found a solution to the problem". The subject-verb-object triplet has been marked with a red border.

Automatic parsing has the disadvantage of being inaccurate, especially in the case of complex sentences, but for purposes such as ours, where the amount of text is very large, the effects of such small inaccuracies should be negligible. This is especially true in our case, where we make use of only the basic relationships of subject, verb and object, where the expected error rate is much lower than the error rate in the more complex relationships in the sentence [55].

2.1.2 The Simplified Clause Structure

For the task addressed in this work - that of recognizing meaningful relation patterns in text - we propose a simplified clause model to capture the elements relevant to the type of pattern we are searching for. The clause model we describe consists of representing whole clauses of text in the form of a *Subject-Verb-Object* triplet. In other words, each clause is represented by the subject of the clause, its verb, and the direct object of that verb. These units are represented by MINIPAR as the surface subject relation (designated "*s*"), the verb relation (designated "*i*"), and the object relation (designated "*obj*"). In fact, we are extracting the top "triangle" in the parsing tree.

We also take advantage of another feature of the parser. When it is able to, the parser produces the root form of a word (un-inflected verbs, or singular form of nouns). These are used (instead of the original form) when provided by the parser, to reduce diversity. A body of text can now be viewed as a series of *Subject-Verb-Object* triplets, containing a summarized and filtered version of the information in the text.

Our clause model is designed to capture the parts of the data most relevant to the actions described in the text. This allows an automated system to filter out the less relevant data (adjectives, adverbs, determiners etc.) and focus on the question of "What is happening in the world described in the text?" The clause model also converts different sentence structures to a single and simplified one.

2.2 The Clustering

2.2.1 Clustering Methods

Clustering is a tool which is widely used in many tasks, and in natural language tasks in particular. The fundamental requirement when using any form of clustering is some way of measuring similarity or, conversely, distance between any two points. Usually, the points are represented in some high-dimensional space, with each point having its own feature vector. The similarity measure is then some function of the two points' vectors. Clusters can be 'hard', meaning that each point belongs to exactly one cluster, or 'soft', meaning that each point belongs to all clusters, with a varying degree of membership. The clustering can be hierarchal, where the larger clusters are made up of a group of smaller cluster, resulting in several layers of partitioning of the data with increasing specificity, or 'flat', where all clusters are independent of one another and do not intersect.

The problem of finding a good feature set and distance measure is a difficult one. Choices are often made after trial and error, or rely on some kind of 'instinctive' knowledge of the data, which is not always clearly defined. Often the feature set is constructed using some information about the data (as in Lin's case), but the distance measure applied is some mathematical measure for vectors (see section 1.3.3). The main advantage of the *Information Bottleneck principle* which we use is that both the feature space and the distance measure are implicitly defined once the target task is decided upon. All that is needed is to define the variable we wish to cluster and the relationship between it and the target variable, the rest follows automatically. The distance function uses the mutual information measure and is defined only in terms of the clustering task, as formulated in the *IB principle*, and is therefore independent of the type of data we use. This is especially important in our work, where the procedure we describe is intended to be suitable for any type of structured sequential data. We cannot therefore rely on an external feature set or distance measure tailored specifically to the data.

2.3 The Information Bottleneck Concept

In this work, we use a clustering method based on the *Information Bottleneck (IB) Variational Principle* formulated by Tishby, Pereira and Bialek [56]. This principle (in its basic form) proposes a method of clustering one variable *X* via its mutual information with another variable *Y* or, in information-theoretic notation $I(X;Y)$. The target variable *Y* occurs in the data together with *X*. For example, if we have a list of movies (values of *X*) and a fixed group of reviewers (values of *Y*), we can present a co-occurrence matrix *M* with a value $M_{x,y}=1\text{-}5$ according to how much each reviewer liked each movie. This data allows us to cluster movies into groups based on the preferences of the reviewers.

The clustering task is viewed as the task of representing *X* through a compressed variable *T*, which has a smaller number of values than *X* while still retaining as much mutual information on *Y* as possible. This task is represented formally as finding the variable *T* which minimizes the *IB functional*

$L = I(T;X) - \beta \cdot I(T;Y)$. The first part of the formula, minimizing *I(T;X)*, is in charge of the compression of *T*. It forces *T* too 'throw away' all the information contained in the original variable *X* which is not relevant to the target variable *Y*. The second part of the formula, minimizing $-\beta \cdot I(T;Y)$, i.e. maximizing $\beta \cdot I(T;Y)$, ensures that information about the target variable *Y* is maintained. The parameter β is used to determine the ratio between the compression and the information preserved. In other words, it allows us to define how much information we are willing to sacrifice for each unit of compression, and visa versa. This principle is called the *Information Bottleneck principle* since the mutual information between the original two variables *X* and *Y* is seen as passing through a restricting bottleneck, represented by the compressed variable *T*, which limits the number of values that can be passed through it.

Each value of the bottleneck variable *T* is, in fact, a cluster composed of a (weighted) sum of values of *X*. In our case, we chose to deal with absolute ('hard') assignments, meaning that each value of *X* belongs to exactly one cluster value of *T*, so the weights are one for values that belong to the cluster, and zero for those that don't.

Why is this a good clustering method or, in other words, why do we expect the elements of the clusters to be similar in some fashion? The IB principle tries to maintain a maximal amount of mutual information between *T* and the target variable *Y*, under the restrictions placed by the compression. The best way to maintain this information while decreasing the number of values is by unifying two (or more) values that are distributed similarly with regard to the target variable *Y*. Since these two values 'behave the same way towards *Y*', there is little information to be gained about *Y* by knowing which of these two values is actually present. It is enough to know that one of them has occurred. Therefore, these two values can be regarded as one value for the purpose of obtaining information about *Y*. They will be clustered together as a single value of *T*.

When using the IB principle in its usual form, we may encounter a problem which stems from values of *X* with a very large number of occurrences. Values with such high counts contain a lot of information about the target distribution, since they cover a large portion of the data points (i.e. co-occurrences of a value of *X* and a value of *Y*). The IB principle seeks to preserve this information, and therefore tries to keep these values uncompressed, by assigning them their own value/cluster in *T*, or assigning them only to a cluster which is composed of values which have a *very* similar distribution to their own. In these cases, these values have a tendency to "take-over" a cluster of *T*. What we get is, in fact, a clustering which preserves the information about *Y* as much as possible taking into account 'quantity' as well as 'quality'. When our purpose is in fact to discover the value of *Y* through the values of *T*, this is exactly what we want. When, on the other hand, we use *Y* only as our measure of similarity in *X*, we would like our clustering algorithm to regard all values of *X* as equal, and not give the frequent ones any preference. This is done simply by normalizing the counts of all values of *X* so that they sum up to one. In effect, we are now looking at a matrix composed, not of co-occurrence *counts* of *X* and *Y*, but of the *distribution* of co-occurrences of all the values of *Y* for each value of X.

2.3.1 The Sequential IB Clustering Method

The clustering method we use in this work is the *Sequential Information Bottleneck* algorithm [57], or *sIB*. This method allows the user to predefine the required size of the compression variable *T*. The method returns a 'flat' partition (non-hierarchal). It is iterative and is guaranteed (under some loosely restricted conditions) to converge to a stable solution (i.e. it will reach a state where further iterations will no longer change the solution). This algorithm can be considered as falling under a general clustering framework proposed by Tsujii *et al.* [58], corresponding to the second type of algorithms mentioned, and employing the CLASSIFY operation defined there.

We chose to use 'hard' partitioning, assigning each word to exactly one cluster with probability of one. Despite the fact that many words are polysemic and may therefore belong in more than one cluster, the alternative – 'soft' clustering - would complicate the pattern acquisition task considerably. If we were to use soft clustering, the appearance of a word in a given pattern would have to be interpreted (with varying probability) as belonging to any possible cluster, and in order to calculate the number of observed instances of a pattern, some very complex calculation would be required. In addition, calculating the expected mean and standard deviation would be very difficult.

The sIB algorithm works as follows:

- ***Definitions***:
  - In order to conform to the more common clustering framework, which deals with *maximizing* a target function, an equivalent formulation of the *IB functional* is used: we wish to *maximize* the functional $L_{\max} = I(T;Y) - \beta^{-1} \cdot I(T;X)$. Note that the parameter and

the accompanying minus sign have been transferred to the term involving the information between *X* and *T*, i.e. the term in charge of *compression*.

- When merging two clusters, t_i and t_j, we can define the *merger cost* as $\Delta L_{\max}(t_i, t_j) = L_{\max}^{before} - L_{\max}^{after}$. Since merging reduces mutual information, the merger cost often a positive value, and relative to the amount of information lost. A lower merger cost is better for our target functional. If we represent this value explicitly, we get $\Delta L_{\max}(t_i, t_j) = [p(t_i) + p(t_j)] \cdot d(t_i, t_j)$, where $p(t_i)$, the probability of a cluster, is simply the sum of the probabilities of its component values, and $d(t_i, t_j) \equiv JS_\pi[p(y \mid t_i), p(y \mid t_j)] - \beta^{-1} \cdot JS_\pi[p(x \mid t_i), p(x \mid t_j)]$, represents the "distance" between the two clusters with regard to their relation to the original variables *X* and *Y*. The notation $JS_\pi[p, q]$ represents the *Jensen-Shannon divergence* in which the parameter $\pi = \{\pi_i, \pi_j\} = \left\{ \frac{p(t_i)}{p(t_i) + p(t_j)}, \frac{p(t_j)}{p(t_i) + p(t_j)} \right\}$ is used.

- ***Initialization***: some (possibly random) partition of *X* into *m* clusters.
- ***Iteration***: at each step, the algorithm draws some value $x \in X$ from its current assigned cluster, and represents it as a new singleton cluster. The algorithm then finds $t^{new} = \arg\min_{t \in T} \Delta L_{\max}(\{x\}, t)$, and merges *x* into t^{new}.
- ***Termination***: The algorithm stops when each $x \in X$ was drawn and re-inserted into its original cluster. This means that changing the cluster assignment of any single value of *X* will not increase the value of the target functional.

It should be noted that this algorithm uses a predefined size for the variable *T*, and therefore already restricts the maximal amount of mutual information that can be preserved between *T* and *X*. This restriction actually fulfils the task of the first term in the IB functional - that of compressing the values in *T*. What is left is the task of the second part of the functional - that of preserving the mutual information on *Y*. In order to cause the algorithm to focus on this task, and not to try to compress *T* any further, we eliminate the second part of the L_{max} functional described above. This is done by setting β to infinity, thereby setting the coefficient of the second part of the functional to zero. This causes the algorithms to ignore the effect of the clustering on the *compression* and focus on preserving the *information*. In this way we are telling the algorithm that the size restriction we imposed is sufficient compression, and we are not interested in compressing T any further.

We chose to ignore the compression/information tradeoff as expressed by β, and handle it ourselves by restricting the number of clusters. This leaves us with the part of the *IB principle* which is of more importance in this work – the definition of similarity between values of the clustered variable as a function of their distribution with regard to the target variable Y. This principle is important because it allows us to perform the clustering without applying a similarity measure external to the data. Instead, it uses the internal structure and the relations within the data.

2.3.2 The Variables and Use of the Clause Model

For the purpose of finding patterns in text, we use *IB*-based clustering to attempt to eliminate "noise" that is irrelevant and disruptive to our task, while preserving those features in the data which are important for pattern recognition. This is, in fact, precisely the task for which the IB principle was formulated! We would like to compress the set of words into a small number of clusters, while maintaining as much information which is relevant to the pattern recognition task as possible.

Unfortunately, the variable of *'pattern recognition quality'* is difficult to formulate, especially in the preliminary stage, where patterns have not yet been recognized. Instead, we use the mutual information of the words with regard to the words with which they have a 'modifier-modified' relationship[1]. In our case, we compress the subject and object words to preserve information about the verbs which they modify, and compress the verbs to preserve information about the subjects and objects which modify them. We view this as a natural and intuitive measure to use in the context of language and sentence structure, and such relations have been commonly used for similar tasks (see sections 1.3.1 and 1.3.3). This method is especially fitting in our case, where we are looking for patterns between actions described in nearby sentences, so relations between components of the action descriptions are very relevant.

[1] Lin and Pantel [59] (and many others) use the complete set of modifier/modified relationships of a word as its feature vector. While this captures a much more detailed description of the word, it also increases the dimensionality of the feature space to almost unmanageable proportions. We chose to use only a very specific subset of these features, namely the subject-verb, verb-object relationships, and only a subset of the possible word values (words which appeared as subject, verb or object more than one-hundred times, see section 3.2).

2.4 The *Clustered-Clause* Representation

Combining our simplified clause model with our clustering algorithm, we now describe the *Clustered Clause* representation. Each *Clustered Clause* is a triplet of clusters. It contains one cluster of subject words, one cluster of verb words, and one cluster of object words. Given an instance of a simplified clause, i.e. a triplet of words which are the subject, verb and object of a clause in the text, we can now assign this triplet to the correct clustered clause by finding the clusters to which the triplet's subject, verb and object belong. These define the clustered clause to which this instance belongs.

Our text data is now transformed into a series of clustered clauses, and this enables us to perform the pattern recognition task on a much 'smoother' set of data, in which many features which were disruptive and irrelevant to the task have been eliminated, such as style and sentence structure differences, and the use of different words with similar meaning. Clustering also helps us to treat many similar words as one unit, thereby increasing the number of occurrences of each component unit of our patterns.

2.5 Pattern Definition

In this work we define several different *pattern templates* for which we search. The first of these represents interactions between the three components of our simplified clause: subject, verb and object. We look for patterns which fit the template "Component. Cluster *i* is likely to appear with Component Cluster *j* in the same clause". For instance, the pattern "Subject Cluster 127 appears with Verb Cluster 22 in the same clause" is an example of a pattern conforming to such a template. Subject Cluster 127 contains subjects such as countries and large administrative bodies, and Verb Cluster 22 contains verbs such as 'maintain' and 'establish'. This example pattern is in fact one of the patterns our system has detected as having high statistical significance. Other patterns recognized by this template can be found in Appendix B.

The second pattern template moves one step up in the data, breaking out of the single clause and into the relations between different clauses in the text. This template is of the form "if Category Cluster *i* appears, Category Cluster *j* is likely to appear within the following *t* clauses". For example, the pattern "if Verb Cluster 5 (verbs such as 'seize', 'attack' and 'invade') appears, Object Cluster 152 (objects such as 'village', 'stronghold' and 'camp') is likely to appear within the following 3 clauses" belongs to this template. This pattern and other patterns recognized by this template can be found in Appendix C.

This template describes an ordered relationship between two different parts of the text, which are connected by the fact that one is closely followed by the other. If the text describes a temporal sequence, these two events may be cause and effect, or possibly two effects of the same cause. This template uses the clustering effect to reduce noise and increase the possibility that instances containing rare words will still be detected. It also makes use of the clause triplet structure by using the component categories when searching for the pattern. It does not yet make use of the clause as a whole unit.

The third pattern template is the most complex, and takes into account the clause structure as a whole. This template attempts to capture the relationship between two whole clauses which are close together in the text. The form of the template is "if Clustered Clause *i* appears, Clustered Clause *j* is likely to appear within the following *t* clauses". Considering the large amount of possible clustered clauses (see section3.3), there are an enormous number of patterns which might fit this template. Naively searching the data for every possible pattern which conforms to this templates is difficult. We therefore imposed a restriction on this template. We require that the clauses be connected on another level – that there be an anchor *word* which is identical in both clauses. The anchoring possibilities we considered were that the clauses shared the same noun (which could be either the subject or the object), or the same verb.

Aside from the fact that anchoring makes the pattern recognition task much simpler, patterns with anchors were considered with certain concrete applications in mind. The Query Enhancement and Entailment tasks mentioned in parts 1.4.3 and 1.4.4 are often of an anchored nature. For example, questions like "Who is the president of the United States?" may rely on patterns of the form "*X* gave an inaugural speech" → "*X* is president". In these cases, it is essential to know, not only which clauses co-occur with each other, but also that these clauses directly involve the same entities, so we can narrow down the number of patterns we search for, and match the relevant entity. The anchored patterns are also more suited to the Anaphora Resolution task, as described in section 1.4.5. The verb anchoring is less suitable for these tasks, and is expected to produce relations of the type VerbOcean deals with (see section 1.3.6), but with the additional fine tuning provided by the subject and object clusters which are the main verb arguments.

2.6 Evaluation Method

We evaluate the patterns our procedure obtained using the statistical *p-value* measure. We count the number of times we observed an instance of our pattern in the data. We then calculate the probability of observing this number of occurrences under the assumption that the two parts of the pattern have no connection between them, or, more formally stated, that they were distributed independently. If this probability is less than 5%, we consider this pattern significant. We chose to use this statistical measure largely on account of its intuitiveness. The notion that if no relation existed between a pattern's components we would expect to see very few instances of the pattern, and the fact that we see many instances indicates the existence of such a relation is a simple one and "makes sense". Other statistical or data-mining evaluation methods could have been chosen instead, and we discuss one such choice in the final chapter.

3 The Procedure

3.1 The Data

For our work in this paper, we used the entire Reuters Corpus (English, release 2000), containing 800,000 news articles collected uniformly from 20/8/1996 to 19/8/1997. This corpus was parsed using Lin's parser MINIPAR (see section 2.1.1). The clause triplets were extracted from the parser's output tree structure. 3,792,956 clause triplets were recovered in this manner.

3.2 Preprocessing

Once we had obtained the list of simplified clause triplets, several preprocessing stages were performed to prepare it for the clustering. First of all, we had too many words for each of the Subject (85,563), Verb (4,593) and Object (74,842) grammatical categories. Many of the subject and object words were proper nouns, and many of these occurred very rarely and were of little interest in the pattern recognition task. In addition, very rare verbs were also less useful, and in some cases were the result of parser misclassification. In order to avoid these cases, and reduce the number of words we handle, we removed the less frequent words - words which appeared in their category less than one hundred times. We then filtered the clause triplets, removing any *triplets* containing those words. This process reduced the number of words in each category dramatically, but only reduced the amount of triplets to 2,933,269, 77% of the original number. The removal of a large portion of the triplets had a smaller effect on the pattern detection task than might be expected. On average, one out of every four clauses was removed, but we must take into account the fact that the few appearances of rare words (especially proper nouns) tend to occur in proximity, so articles containing them had disproportionately large pieces removed, while most other articles were much less affected.

The second preprocessing step involved the removal of all words (and the clause triplets that contained them) that were one letter in length, other than the word *'I'* or a single digit. These were mostly initials in names of people or companies, which were uninformative to us in the clustering task, since there was no way to separate letters that represented different people with the same initial. This processing step brought us to the final count of 2,874,763 clause triplets (75.8% of the original number), containing 3,153 distinct subjects, 1,716 distinct verbs, and 3,312 distinct objects.

3.3 Clustering

The clustering was done in two stages. In the first stage, all subjects were clustered into 200 clusters using their co-occurrence matrix with the verbs. The same was done with the objects. We now had 200 clusters of subjects and 200 clusters of objects. In preparation for the next stage, we define a new variable, for which the values were all the possible pairs of a subject cluster and an object cluster. In our case, there were 200x200 = 40,000 such pairs possible. We now produced a co-occurrence matrix where the rows represented the verbs, and the columns represented pairs of subject - object clusters. In other words, the cell [*i, j*] in the matrix held the number of times the verb *i* occurred in the same clause with a subject and an object belonging to the subject and object clusters which make up the pair *j*. We used this co-occurrence matrix to cluster the verbs into 100 clusters.

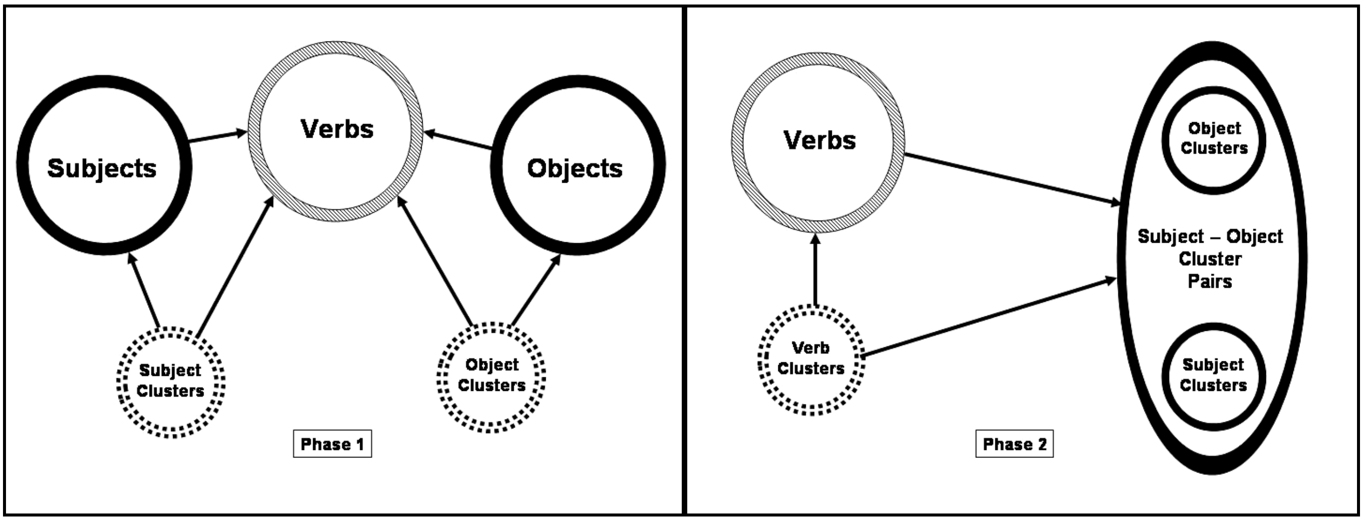

Figure 2 The two clustering phases. The target variables are marked with a lighter border, the variables to be clustered with a darker one, and the clustered variables with dotted outlines. The arrows represent the dependencies between the variables which we use in the clustering.

At the end of the clustering procedure, we had 200 subject clusters, 100 verb clusters, and 200 object clusters, and knew how to assign each word in the filtered data file to its cluster. From these numbers, it is easy to see that there were 200x100x200 = 4 million possible clustered clauses. In fact, in the data, only 8% of the possible clauses actually occurred; see section 4.1.3 for details.

3.4 Simple Pattern Detection

For the purpose of detecting the simple patterns comprised of a pair of category clusters (templates of the first and second type described in section 2.5), it was possible to hold in memory a two-dimensional table of counts, where the rows represented the first half of the pattern (all possible values of a certain category cluster) and the columns represented the second half (all possible values of another category cluster). For example, if the template we were searching for was "Subject Cluster *i* appears with Verb Cluster *j* in the same clause", the cell (23, 15) in the matrix held the number of instances where Subject Cluster 23 appeared with Verb Cluster 15 in the same clause.

A program then went over all the clustered clauses in the filtered data file, and counted the instances of each pattern. This procedure was repeated separately for each possible combination of two grammatical categories. For templates of the first type described above, which deal with the same clause, there are only three possible combinations (subject-verb, subject-object and verb-object), whereas for templates of the second type, there are nine, since both parts of the template can be of the same category, and when we deal with two different categories, there is significance to the order of the two parts. In templates of the second type, the method was very similar to the one used for the first type, except for the fact that the program had to look *t* clauses ahead to check for co-occurrence, according to the specified *t*. We used *t* = 3, 6, 9 and our results show that the amount of new patterns detected steadily declines as we approach the higher values (see section 4.3.2).

After obtaining this matrix of counts, we can now ask how many instances of each pattern were observed in the data. We then compare this to the number of instances expected if we assume the two parts of the pattern are distributed independently, and calculate the statistical significance of the pattern from these values.

3.5 Complex Pattern Detection

In order to detect instances of the third template type, more sophisticated methods were needed. It was not feasible to maintain a table where there was a row for each possible value of the first part of the template, and a column for every possible value of the second part, since each part was actually a clustered clause, which means each part had 4 million possible values in our setting (even the actual number of values occurring in our data – more than 320,000, was too high). Instead, we used the anchor words to assist in the task, utilizing them as keys in a lookup table. For each clustered clause we read, we treated that clause as a possible left half of the pattern, and put it in the hash table with the anchor as the key. We also searched the table of existing candidate left parts to see if the current clustered clause could complete an instance of a pattern or, in other words, whether it matched the right half of any template that had a left half already present in the table. In addition, we required that the matching left part we found had appeared at most *t* clauses ago.

3.6 Reducing Generalized Patterns to Specific Ones

Before calculations were made to determine which patterns were significant, another processing step was performed. In some cases, the patterns we found were unnecessarily generalized. For instance, in patterns of the first type we assume that the pattern is a result of the connection between two category clusters, but it is possible that in many, or perhaps even in all, of the instances of that pattern which we saw in the data, only a single word appeared as the representative of one of the clusters composing the pattern. For example, one of the verb-object patterns detected for the first template type was "Verb Cluster 9 is likely to appear with Object Cluster 129 in the same clause", but all the instances of this pattern actually observed in the data were the phrase "breathed a sigh (of relief)".

In cases like this, our pattern is over-generalized, since it assumes that any word from the cluster can appear in that position in the pattern, when, in fact, we have significant observations of only a single word. In the example above, our pattern should really be of the form "the verb 'breathe' is likely to appear with the object 'sigh' in the same clause" (as can be seen in Appendix B, subsection 4). Before proceeding to score the patterns by their statistical significance, the program went over each one to check whether the over-generalization phenomenon occurred in the pattern, and if so, replace that pattern with the more specific version.

It should be noted that the significance calculation is also different for the more-specific clusters. Instead of considering the probability of co-occurrence of two clusters under the independence assumption, we are now considering the probability of the co-occurrence of a word and a cluster (or in some cases, of two words) which is much lower, and raises the significance of an observed high count.

3.7 Significance Calculation

In order to calculate the statistical significance of each pattern, it is necessary to know the number of occurrences expected to appear in the data under the assumption that there is no connection between the parts of the pattern or, in other words, each part is distributed independently. For the first type of pattern template – the one composed of two different category clusters occurring together in the same clause, it is simply a matter of counting the occurrences in the data of each one of the component clusters. These counts represent the distribution of the clusters in each category. We then multiply these probabilities to get the probability of the pair occurring in the same clause under the independence assumption.

Given the independent probability of the co-occurrence, we can use a Gaussian approximation of the binomial distribution with a mean of $\mu = n \cdot p$, and a standard deviation of $\sigma = \sqrt{n \cdot p \cdot (1-p)}$, where n is the total number of clause triplets in our data and p the probability under the independence assumption. Using these parameters, we wish to find a threshold k such that the probability of observing k occurrences or more, under the independence assumption, is less than 5%. For a Gaussian distribution with a mean and standard deviation such as described above, the threshold k is located at a distance of two standard deviations away from the mean. So, for our purposes, if the number of observed occurrences is more than $k = n \cdot p + 2 \cdot \sqrt{n \cdot p \cdot (1-p)}$, the pattern is significant.

For the second type of pattern template, the significance calculation is similar, but the calculation of the mean and standard deviation of the pattern under the independence assumption is much more complicated[2]. In order to estimate these values, we generated random sequences of pairs according to the cluster *pair* distributions we observed in the data (this pattern template assumes independence only between one clause and the next, not within the clause itself). For instance, if we are looking for a subject-verb pattern, we generate a sequence of subject-verb pairs (these fill the part of clauses, since the object part is not needed) according to the distribution of such pairs in the actual data. We then calculated the mean and standard deviation over 100 repeats of this simulation. For a given pair of categories, for instance subject-verb, this simulation system provided two tables, one containing the expected means, and the other the standard deviations of each possible pair of subject and verb clusters. These tables were used to evaluate the significance of the pattern we found in the actual data, as in the previous pattern template (a distance of two standard deviations from the mean was considered significant). These tables need to be calculated separately for every value of *t* we consider.

For the third type of pattern, the problem lies once again with the task of calculating the probability of the pattern occurring under the assumption of independence. In this case, the two parts of the pattern are not single clusters, but clustered clauses. As you may recall, there is one position in each part of the pattern which contains the anchor. If we ignore the anchor position for the moment, and its influence on the co-occurrence probability, we are left with a pattern template composed of two parts, each of which is a pair of category clusters. If we could calculate the means and standard deviations for each possible pattern matching this template, as in the tables described above, the calculation would be identical to the one we used for the second template type in the previous paragraph. Unfortunately tables containing the expected mean and standard deviation of co-occurrences of every combination of two cluster pairs (which comprise the two parts of the pattern) would be extremely large. Instead, we built the tables for certain specific cluster pair probabilities $p = 1\cdot10^{-7}, 2\cdot10^{-7}, 3\cdot10^{-7}, ...,$ $9\cdot10^{-7}, 1\cdot10^{-6}, 2\cdot10^{-6}, 3\cdot10^{-6}$, etc. and used these values for the calculations. This was done as follows: if we wanted to know the expected mean and standard deviation of a co-occurrence of cluster pair A and cluster pair B, we looked up the probability of encountering each of the cluster pairs, lets say p_A = 0.0034 and p_B = 0.00086, we rounded them *up* to the most significant digit, in this case 0.004 and 0.0009, and looked up the mean and standard deviation in the tables. By always rounding up, we ensure that we only increase the expected number of counts, thereby raising the significance threshold.

[2] Also, it is less clear that the distribution behaves similarly to a Gaussian distribution, though R. Gwadera et. al [60] have shown this to be true for an almost identical case, based on Theorem 27.5 of [61] , which applies to our case as well.

We now return to the issue of the influence of the anchor on the expected pattern probability. We would like to take into account the probability that the anchor word appeared with the cluster pair comprising the rest of the clause. We would also like to take into account the probability that the anchor word was repeated (in two specific parts of the clauses) within *t* clauses. Unfortunately, this probability depends on the anchor word and the cluster pairs, and is quite complicated to calculate. It will probably also be quite a rare event, in which case the simple method of counting the occurrences within the data, which works well for clusters or even cluster pairs, will not provide a good assessment. For these reasons, we decided to perform the significance calculations without considering the anchor influence. Since the anchoring will only reduce the probability we use to predict the expected number of pattern occurrences, ignoring it simply raises the number of observed occurrences we require to pass the significance test. By ignoring the influence of the anchor, we are simply making it harder to find patterns, and raising the significance of the ones that we *do* find.

4 Results

In this chapter we present the results of the various tasks we set ourselves at the beginning of this work. We show the relationship patterns detected by our methods for each of the three template types. The generalized patterns resulting from the second template type constitute a new type of relation, previously only considered for verbs (see section 1.3.6), and in a non-generalized form. Those matching the third template type represent the results of our approach to capturing generalized relations involving whole clauses, a complex problem which has not been adequately addressed in the past.

Besides the relationships we detected, our results should be of interest for other reasons. Those working in the fields of machine learning or data mining should be interested in the fact that clustering using mutual information within the same time-frame (or sentence, in our case) provides clusters of high objective quality (section 4.14.1.2), and that these clusters are suitable for discovering patterns between different time-frames (sections 4.2 and 4.3). We also show that it would have been impossible to obtain these results using non-clustered words (section 4.5).

For those interested in the linguistic aspects there are many phenomena within the data which could be of interest, for instance the amount of information each component of the clause holds about the others (section 4.1.1), and the similarity between manually grouped words and automatically clustered ones (section 4.1.2). Another interesting phenomenon is the high amount of connectivity between different clauses and the influence of each of the clause components in this relation (sections 4.3 and 4.4). Also, the way in which word clusters are repeated (section 4.3) should be of interest.

Before examining the results in detail, a certain data-specific factor should be taken into account. The Reuters corpus contains detailed berthing schedules of ships at different ports. These schedules are lists of ship's cargos (such as "Containers 398/52 ID Orient Mumbai 27/08", "Wheat flour/rice 246/-- ID Eco Ekram 25/08"), berthing times, delays, etc. Since these lists are not properly formed clauses, the parser failed to correctly annotate them, and, looking desperately for verbs, interpreted nouns as verbs if there was any such possibility. We therefore have verb clusters 56 and 68 (see Appendix A). Also, we have many clauses where 'vessel' is the subject or object, and "verbs" from the two clusters we mentioned are the verb.

4.1 Clustering Results

4.1.1 The Clusters

The full list of clusters and their component words can be found in appendix A. It is quite clear that in most cases the words in each cluster are related in some fashion, either through a similar meaning or by belonging to some unifying group. For example, Verb Cluster 6 (verbs signifying destruction) and Subject Cluster 193 (different types of commodities or raw materials).

In some cases, however, one may notice that a cluster seems to be composed of more than one clear sub-group (see, for instance, Object Cluster 20). This is the result of the artificial restriction we imposed on the number of clusters. The clustering algorithm preferred to perform a more informative separation on a different cluster, at the expense of separating this one. In light of this, it is interesting to note the "fine-tuning" of some of the clusters, for instance, the fact that we have a cluster of countries involved in military conflicts (Object Cluster 3), and another for other countries (Object Cluster 42), a cluster for winning game scores (Object Cluster 166) and another for ties (Object Cluster 195), etc. The fact that the algorithm separated these clusters indicates that the distinction between them is important with regard to the interactions within the clause. This is clear, for instance, in the first example above - the context in which countries from the first cluster appear is very different from that involving countries from the second cluster.

Examining the clusters, it is easy to notice the effect of the dependencies we used when constructing the clusters. Many clusters can be described by labels such as "things that are thrown" (Object Cluster 70), or "verbs concerning attacks" (Verb Cluster 75). While such criteria may not be the first choice of someone who is asked to cluster verbs or nouns, they represent unifying themes which are very appropriate for our pattern search. We wish to detect connections between actions described in the clauses. For instance, between throwing and military/police action (most of the throwing described in the news reports fits this relation). In order to do this, we must have clusters which unite the words relevant to those actions. Other criteria for clustering would most likely not be suitable, since it would probably not put 'egg', bottle' and 'rock' in the same category. The clusters we produced, on the other hand, allowed us to detect this relation using the second template type - the pattern "OC_70 –(3)- VC_1".

We will now display and discuss the clusters from an information-theoretic point of view. The following figures show the amount of information preserved between the clustered variable T and the target variable Y, as a percentage of the original mutual information between the unclustered variable X and Y. As can be seen from the figures, the compression/information curves of the subject and object categories behave very similarly, rising steeply until we reach approximately 1000 clusters (three words in each cluster, on average), and then gradually flattening out. The curve for verbs has a similar general behavior, but, since there were only 1,700 values to begin with, starts to flatten out at about 500 clusters. We chose the number of clusters in each category that preserves approximately 50% of the original mutual information (see Table 1 below). More information would have been preserved if we had used a larger number of clusters, but at the expense of increasing the dimensions of our clause representation.

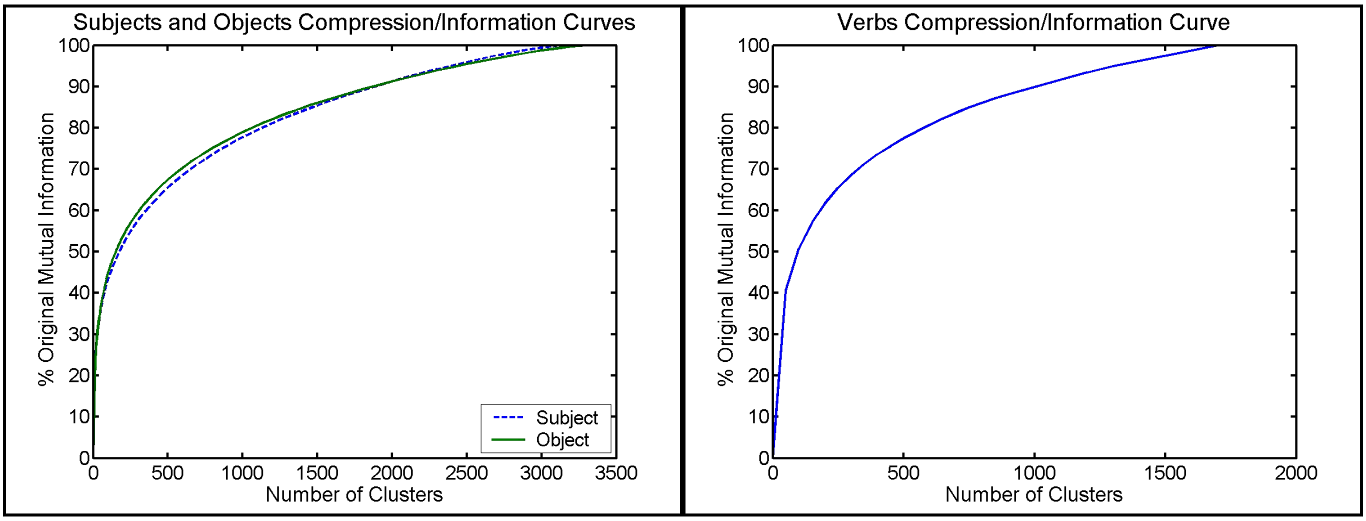

Figure 3 Amount of mutual information preserved by the clustered variable as a function of the number of clusters. The first figure shows the curves for the clustering of subjects and objects (Phase 1, see section 3.3), and the second shows the curve when clustering verbs (Phase 2).

The following table lists the amounts of original and preserved mutual information[3] in each of the categories in our representation (all entropy and mutual information quantities are in bits).

[3] Since we use the normalized version of the co-occurrence matrix (see section 2.2) the mutual information shown here is not the mutual information between the two variables, but rather the mutual information between a normalized variable (which we want to cluster), where all the values have the same probability, and an un-normalized one. This is why the mutual information between subjects and objects is different from the mutual information between objects and subjects, even though mutual information is a symmetric function.

	Entropy of the Variable to Be Clustered	Original Amount of Mutual Information	Amount of Mutual Information Preserved After Clustering (*)
Subjects Clustered by Verb	11.62	3.4 (29.2% of the entropy)	1.76 (51.7% of original M.I.)
Objects Clustered by Verb	11.69	3.74 (32% of the entropy)	2.0 (53.5% of original M.I.)
Verbs Clustered by Subject and Object Clusters	10.75	5.52 (51.4% of the entropy)	2.8 (50.7 %of original M.I.)
Subjects and Objects	See Above	Normalized Subjects - 4.19 (36%) Normalized Objects – 3.44 (29%)	Clustered Subjects – 1.97 (47%) Clustered Objects – 1.66 (48%)

Table 1 Mutual information quantities between the clustered and unclustered variables in the clause representation.

As seen in the table, the mutual information between subjects and objects is greater than that between either one of these and verbs. Despite this, the percentage of information preserved in the clustering between subjects or objects and verbs is greater; indicating that verbs provide a better target variable when clustering these categories. As stated before, we are not interested in predicting the exact values of the target variable from the clusters they appear with, but rather in capturing some inherent similarity between the clustered words (this is also the reason we normalized the values). The higher percentage of preserved mutual information indicates that there is more similarity between each cluster's component words if we use verbs.

4.1.2 Evaluating the Quality of the Clustering

While the similarity between the words in the clusters may be clear to a human reader, it is important to quantify this similarity in an objective manner. We claimed that our clustering method, based on the clause structure as described in section 2.3.2 and 3.3, captures an inherent similarity between the words that is relevant for more than just the relationships by which the clustering was performed. In order to demonstrate this, we used WordNet [36] and the WordNet–Similarity Package [62], to calculate the average pair-wise distance between words in the same cluster, and compared it to the average pair-wise distance between words in randomly generated clusters. The WordNet-Similarity Package allows the use of several distance measures (see [22] for a detailed overview). We chose to show the results of the Resnik [27], the Leacock & Chodrow [28] and the Hirst-St.Onge [29] measures, since they are based on the WordNet graph structure, which was manually constructed, and not on co-occurrence information.

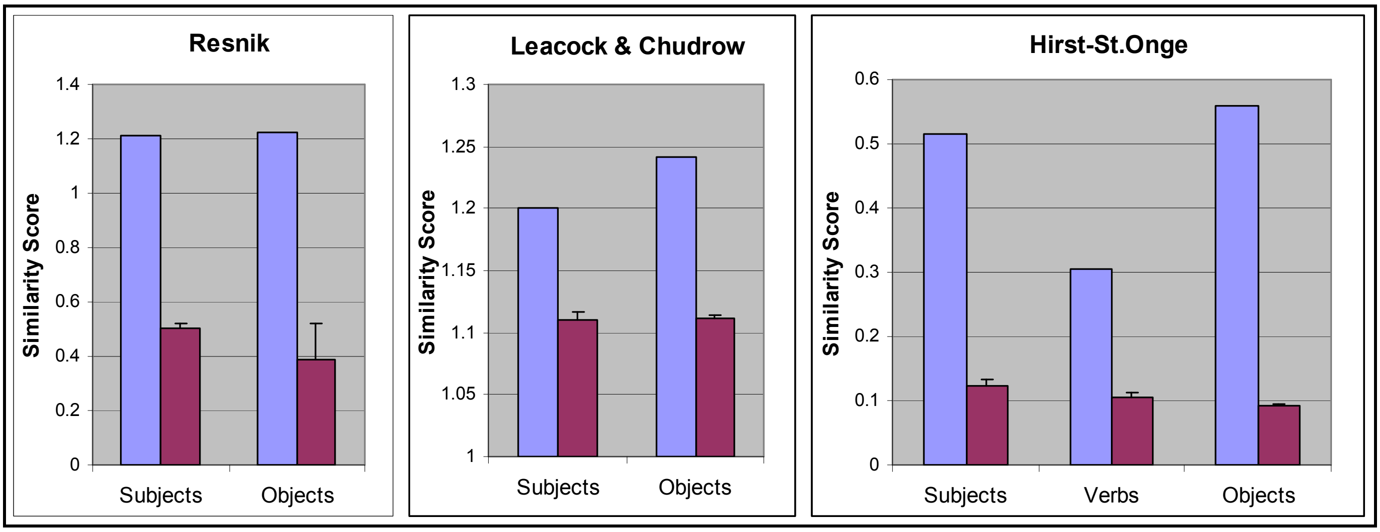

Figure 4 Average pairwise similarity within clusters for our clusters (left) and random clusters (right). The similarity score for the random clusters is the average over ten random clusterings, and the standard deviation is shown. The first two similarity measures are defined only for nouns.

As we can see in the figure, the pair-wise similarity is far greater in our clusters than in any random clustering, for both similarity measures. This was true for the other similarity measures in the package as well.

Another way to show the relevance of our clustering to other tasks is information based. In our procedure, we used the subject-verb and object-verb relationships to cluster subjects and objects. Once we have the produced the clusters, we can see how well they preserve mutual information with regard to relationships other than the ones we used. If the clustering results were only good for the task of preserving information between the two variables we designated when performing the clustering, we would expect these clusters to perform little better than random clusters in the preservation of mutual information in relationships other than the one we used. On the other hand, if the clustering has indeed captured a "natural" grouping of the values, there should be an inherent similarity between words in the same cluster, which would also be relevant to other relationships. As we can see in the figure below, the latter case is what actually occurs.

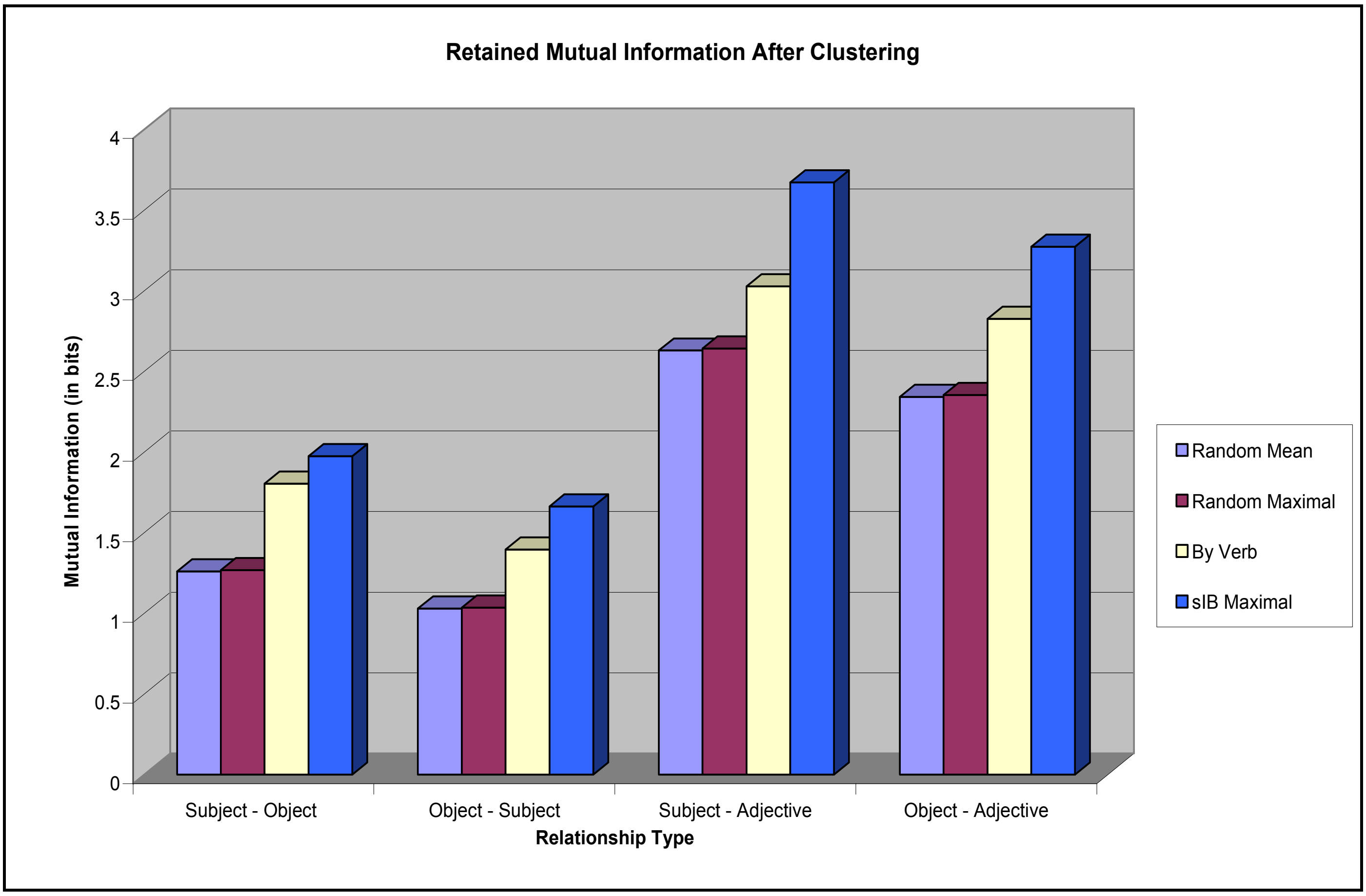

Figure 5 In the figure we see the amount of mutual information retained by the clustered variable (first in each pair) with regard to the target variable, in four different relationships. The relationships involving adjectives refer to adjectives in the clause which describe the clustered variable. The leftmost column shows the average amount of mutual information that was preserved when 100 random clusterings were performed (the standard deviation was less than 0.2% in all cases). The next column shows the highest mutual information obtained in those 100 random trials. The third column shows the amount of mutual information preserved when we use the clusters obtained by the sIB algorithm when using *verbs* as the target variable (these are the clusters presented in Appendix A), disregarding the actual target in the relationship. The rightmost column shows the maximal amount of mutual information between the two variables in the relationship, as obtained by using the sIB algorithm (described in section 2.3.1) with that target relationship. As we can see, the verb clustering performed much better than random, even for very different relationships.

4.1.3 The Clustered Clauses

In the filtered data, only 320,725 distinct clustered clauses occurred. This represents 8% of the four million clustered clauses theoretically possible in our setup. The average number of appearances of a clustered clause which *did* appear in the data is 8.96. The highest number of occurrences of a clustered clause was 23,638, for the clustered clause "Subject Cluster 51, Verb Cluster 46, Object Cluster 75". A look at the words in these clusters gives the explanation – Verb Cluster 46 contains the verbs 'tell' and 'inform', and Object Cluster 75 contains the word 'Reuters' (Subject Cluster 51 is composed of important officials). The appearance of only 8% of the possible clustered clauses is the result of the way the clustering was performed. Since words with a similar co-occurrence distribution with regard to the target variable were clustered together, we expect the cluster containing them to act more-or-less like each of its components. This means the clusters are expected to have a specific co-occurrence pattern, i.e. they co-occur mostly with certain words (or in the cluster sense – with the clusters containing those words). These specific co-occurrence patterns mean that many combinations of clusters are very rare, or do not exist at all.

4.2 Intra-Clause Patterns

The most statistically significant patterns found for the first template type (cluster co-occurrence within the same clause) are presented in Appendix B. These are partial results, listing only the 50 most significant cluster relations, due to lack of space. We also list a few of the word-word relationships within the clause, which are relevant to the discussion. The following table shows the numbers and composition of the results.

	Number of Possible Relations	Number of Significant Relations	Number of Cluster – Cluster Relations	Number of Word – Cluster Relations	Number of Word – Word Relations
Subject - Verb	20,000	11,536	11,168	252	116
Verb – Object	20,000	12,475	12,147	229	99
Subject - Object	40,000	20,901	20,020	607	274

Table 2 Numbers and composition of significant intra-clause patterns.

It is interesting to note that, in the case of subject-verb and verb-object cluster interactions, more than half of the possible interactions are in fact significant, indicating that there is a strong correlation between the verb and its subject, and between the verb and its object. On the other hand, there are a smaller percentage of significant interactions between subject and object. The clustering was done to preserve the subject-object and verb-object mutual information, but did not consider the subject-object mutual information. Despite this, we do see many significant subject-object interactions, indicating that the clustering we performed on the subjects and objects according to verbs is also suitable for capturing interactions of other kinds (as seen in section 4.1.2).

A close look at the results shows that in the word-word and word-cluster patterns, there are several cases where the co-occurrence is a clear case of a (consistent) parser error. For example "red - chip" where 'red' was tagged as the subject of the clause, and 'chip' as the verb. In other cases, the connection is a result of some linguistic expression, such as "breathe - sigh", and "pour - scorn". Some cases represent an artifact specific to the dataset we use, as in the "000s - except" which is the result of the sentence "All Data Above 000s Except Per Share Numbers", which appears in many Reuters financial news reports.

If we accept the claim proposed in this paper – that the clustering captures a natural grouping of the words, and is relevant to the kind of interactions in which we are interested – we can conclude that word-word (or word-cluster) co-occurrence without a more general cluster-cluster relationship represents an unusual event and requires further examination. It may be due to one of the factors mentioned above, and, if so, we may decide to ignore it in the pattern search. This phenomenon may also provide a method to validate parser accuracy and has implications regarding the resolution of word-sense ambiguities[4].

[4] If a word was tagged as a verb, but doesn't behave in a fashion similar to other verbs with the same feature distribution, it is likely that the tagging was incorrect. See also McCarthy *et al.* [30] who used cluster conformity (or, in this case, nearest-neighbors conformity) for this task.

4.3 Inter-Clause Patterns

4.3.1 Patterns within Three Clauses (t = 3)

The most significant inter-clause results are listed in appendix C. In the inter-clause patterns there were much fewer word-cluster and word-word patterns. It is possible to see that in these results as well, we have retrieved patterns which are the result of parser errors (most noticeably verb clusters 68 and 56, see final paragraph in section 4). We also see some examples of patterns which are an artifact of the data we use, such as "SC_64 -(3)- SC_79" (the first of the "SC-(3)-SC" patterns), which is, once again, the result of the Reuters financial reports, which contains many statements such as "0830 MON U.K. EX FD DRNK Y" which are not properly formed sentences.

The following table displays the number of patterns found for the second type of template, with *t* =3, for each possible pairing of categories. The percentage of patterns found from the total number of possible patterns of this type is shown in parentheses. The rows represent the category appearing in the first half of the pattern, and the columns represent the category appearing in the second half.

	Subject	Verb	Object
Subject	6068 (15% of possible)	3566 (17.8% of possible)	6667 (16.7% of possible)
Verb	3612 (18% of possible)	1974 (19.7% of possible)	3828 (19.4% of possible)
Object	6722 (16.8% of possible)	3913 (19.5% of possible)	6875 (17.2% of possible)

Table 3 Number of significant patterns found for pattern template II, with *t* = 3

It is interesting to note that the percentage of significant patterns is similar for all category combinations. Despite this similarity, it is obvious that patterns where one of the components is a verb have a larger percentage of significant instances, and that the patterns connecting verb to verb show the highest percentage of significant instances. It is also interesting to note that there is a larger percent of connections involving objects than of those involving subjects. This seems to indicate that verbs play the strongest part in the continuity of the text, then objects, with the subjects providing the weakest connections. The continuity we refer to is of a special kind – we are talking about connections between clusters. It is likely that there are many connections (mostly repeats, as described below) between nouns in nearby sentences, but as long as they are in the same clusters, we only count one connecting pattern. What we observe here is that there is a strong continuity between *different* verb clusters – in other words, verbs describing different actions are connected in the sequence of clauses. This is what we are looking for when we search for cause-and-effect patterns.

Another interesting (though not surprising) fact, is that repeats of the same cluster are the most significant type of pattern. Approximately 90% of the possible patterns composed of a repeat of the same cluster appear significantly in the text. It is, of course, of interest to see which patterns of this kind do *not* occur, and why. We can speculate that these clusters contain words with similar meaning but which are used in different contexts, so that they are unlikely to occur in the same group of sentences. Another interesting fact about the repeating patterns is that while in most instances of the pattern, the repeats are actually of the same word, there are also many instances where repeats of the cluster were represented by different words within the cluster. For instance, in the pattern "VC_40 (within 3 clauses) VC_40", there were many cases where the one of the component words were repeated within three clauses, but also many cases where the words were different, such as 'detain' followed within three clauses by 'arrest', or 'massacre' followed within three clauses by 'kill'. Another example, in the noun patterns, is the pattern "SC_39 (within 3 clauses) SC_39", where again, most of the instances were repeats of the same word, but there were also instances such as 'carmaker' or 'automaker' followed within three clauses by 'Mercedes', 'Honda' or 'Nissan'. Another interesting case is the pattern "SC_99 (within 3 clauses) SC_99", where we find twice as many instances of 'explosion' followed (within three clauses) by 'blast', than the other way around, indicating that order is in fact of importance in these types of patterns.

Repeats of clusters using *different* words are much more common in verbs than in subjects or objects. We counted the number of repeat patterns where more than 20% of the instances were different words, and found that for verbs, 74% of the repeats met this criterion, whereas for objects only 66% did, and for subjects, even less – 52%. The difference between verbs and nouns is likely to be the result of the fact that there are many ways of expressing the same action using different words, whereas nouns (especially pronouns) have few synonyms. It is harder to explain the difference between subjects and objects.

We mention the repeat and continuity phenomena only as examples of further uses for which our process can be employed. These phenomena warrant further research (for instance – comparison with accepted theories about continuity in text), but this is beyond the scope of this work.

4.3.2 Longer-Range Patterns (t = 6, t = 9)

As mentioned in section 3.4, we searched for significant inter-clause patterns within a distance of three, six and nine clauses. One of the purposes of using multiple distance values was to see if we could find patterns that are specific to a certain distance span. For instance, we would expect clauses describing an immediate response to, or result of, a certain action to be described in close proximity to the clause describing the action. On the other hand, since we are using news texts, we can perhaps expect comments by other people not directly involved to follow further along in the text. Another reason for the different distance measure is for calibration of the system. We would like to know over what distance span we can expect to find significant patterns, and which distances are either too short to catch all the instances of the patterns, or longer than is actually needed.

In the following tables we present the number of new patterns found for $t = 6$ and for $t = 9$. By "new patterns" we mean patterns that weren't found (or were not statistically significant) for a smaller value of t. In other words, the first table presents the number of significant patterns we found for $t = 6$, which were not found for $t = 3$, and in the second table we present the number of significant patterns we found for $t = 9$, but not for $t = 6$. As in the previous section, the rows represent the category appearing in the first half of the pattern, and the columns represent the category appearing in the second half.

	Subject	Verb	Object
Subject	1324 (20 % of 6658 found)	630 (16.7% of 3780 found)	1237 (17.5% of 7078 found)
Verb	641 (17 % of 3786 found)	357 (17 % of 2109 found)	725 (17.6% of 4114 found)
Object	1259 (17.5% of 7181 found)	661 (16 % of 4117 found)	1378 (18.5% of 7423 found)

Table 4 Number of significant new patterns found for *t* = 6, but not for *t* = 3

	Subject	Verb	Object
Subject	985 (14 % of 6981 found)	449 (11.7% of 3851 found)	896 (12.3% of 7273 found)
Verb	463 (12 % of 3870 found)	216 (10.3% of 2103 found)	435 (10.5% of 4161 found)
Object	862 (11.7% of 7361 found)	459 (11% of 4176 found)	868 (11.6% of 7505 found)

Table 5 Number of significant new patterns found for *t* = 9, but not for *t* = 6

As we can see in the tables, in the transition from $t = 3$ to $t = 6$, we increase the number of significant patterns we find by 17-20% (note that the percentages in the table are out of the total number of patterns found for the larger value of t, meaning that the *increase* percentage is higher), whereas moving from $t = 6$ to $t = 9$, we only increase the numbers by 10-14%. As expected, as we increase the distance which the pattern can span, we find more (still significant) patterns, but the number of new patterns decreases.

It is also interesting to note that the largest increase in new patterns is with interactions involving the subjects, with the subject-subject patterns having the strongest increase. What we see here is that in the patterns specific to a longer range, subjects provide the strongest connection, unlike the patterns which also occurred in short-range, where we saw that verbs played this part (see previous section). It is possible to surmise that longer range patterns are connected by the entities that performed the actions, and not necessarily by the actions themselves. If we are, for instance, interested only in direct cause-and-effect patterns, we might want to check for short range patterns. If we are interested in the influence of an action on the entities involved, we might look for patterns specific to longer ranges.

4.4 Complex Patterns

In section 2.5 we described the third type of pattern templates we searched for. This template is the most complex one we deal with. In Appendix D are listed the most statistically significant patterns we found using this template, for each possible anchoring system. Here the influence of Reuters' docking manifests (see end of section 4) is strongly felt. Patterns deriving from these schedules were noted in **bold** font.

The following table lists the number of patterns found, and the composition of the patterns. For each anchoring system, we indicate the number of total patterns found, and how generalized these patterns were. We divide the generalization into two categories:

- **Generalized patterns**: the pattern is composed mostly of clusters (no more than two out of the four cluster positions in the pattern template were represented by a single word in all the instances).
- **Specific patterns:** three or more of the cluster positions were represented in all the instances by a single word

Anchoring System	Total Patterns Found	Generalized Patterns	Specific Patterns
Subject – Subject Anchoring	428	47 (11%)	381 (89%)
Verb – Verb Anchoring	310	92 (29.6%)	218 (70.4%)
Object – Object Anchoring	291	45 (15.5%)	246 (84.5%)
Subject – Object Anchoring	180	28 (15.5%)	152 (84.5%)
Object – Subject Anchoring	178	22 (15.7%)	156 (84.3%)

Table 6 Numbers and compositions of patterns found using template type III.

As can be seen from the table, the different anchoring systems produce very different results. Most notably, when we use verbs as the anchors, the composition of pattern types is very different. When we look at the patterns found, we can see that there is a qualitative difference as well. As may be expected, most of the patterns we find are simply disconnected sentences containing verbs that tend to be repeated in the same context. In reports of military or terrorist action, the verb 'kill' is often repeated, when the entities involved are not necessarily the same. The same is true for financial reports and verbs such as 'buy' or 'sell', and for news reports in general, with the verbs 'say' and 'tell'. Since the link between the pattern parts does not exist through the entities involved, it is hard to know whether the two patterns are related in the cause-and-effect sense or in a more general, contextual sense. On the other hand, the rest of the anchoring systems, which use the entities involved in the action as anchors, produce patterns that are more directly related.

It is also clear from the table that the anchoring between the same categories produces more patterns than between the same noun in different grammatical roles. This is expected, since many nouns can only play a certain part in the clause (for instance, many verbs cannot have an inanimate object as their subject).

Another conclusion we can reach from these numbers is that the third pattern template (using anchors) is coming close to the limit in pattern complexity. Because of the complexity of this type of template, the instances of patterns become rare, and once again we are faced with a problem of data sparseness, although on a much higher level than single words. The number of instances of (generalized) patterns we found for this template was quite low, and it is likely that some patterns were missed simply because, even though the probability of encountering such a pattern is much higher than would be expected under the independence assumption (meaning that the pattern is statistically significant), it is still very small, and therefore instances of the pattern may not occur noticeably in the text. In order to successfully search for even more complex patterns, a solution to this problem must once again be found. One possibility is to cluster whole clauses, but there may be other clustering possibilities, or other solutions altogether.

4.5 The Influence of Clustering

In the final section of this chapter, we would like to discuss the influence of the clustering on the pattern searching task. Our purpose in employing clustering for this task was twofold. First, we were interested in overcoming the problem of data sparseness. We wished to find patterns that contain rare component words, which do not appear frequently enough in the data for us to detect their appearance in patterns. In order to show that clustering helped us with this problem, we will bring a few examples from each type of pattern template, where clustering overcame this problem, and helped detect patterns which we would not have been found otherwise.

1. Intra-Clause Patterns:
   - The pattern "SC_159 - VC_16" was among the significant patterns found (though it was not among the top 50 displayed in Appendix B). The subject cluster contains large companies, while the verb cluster contains actions performed on prices and profits. This is a logical connection, but no subject and object words from these clusters co-occurred in the data more than twice. Two appearances are within the statistical deviation of all but the rarest words, and would not have been detected as significant without the clustering effect, so this pattern would have been overlooked.
   - The pattern "VC_61 - OC_116" was detected as significant (though it too was not among the top 50). The object cluster contains types of relationships and status. The verb cluster contains actions regarding relationships or status, such as 'neglect', 'dislike' and 'admire'. Once again, no two words from these clusters co-occurred in the data more than twice.

2. Inter-Clause Patterns:
   - The pattern "SC_27 –(3)- VC_88" was detected as significant. The subject cluster is composed of activists and people with strong opinions (note the fine-tuned category!), the verb cluster contains likely effects of such opinions. Here too, no specific pair of words from these clusters appeared within three clauses of each other more than twice.
   - The pattern "VC_52 –(3)- VC_20" was also retrieved. The first cluster contains verbs related to military or law enforcement actions, and the second contains actions which are illegal or covert. As in the previous examples, this connection would not have been detected, for the same reasons.

3. Complex Patterns:
   - The pattern "x, C_40, C_165 -(10)- x, C_52, C_152" (fourth in Appendix D, subsection 1) appeared only four times. In these four instances, Verb Cluster 40 was represented twice by the verb 'arrest' and twice by the verb 'detain'.
   - The pattern "C_37, visit, x -(10)- x, C_46, C_63" (eighth in Appendix D, subsection 5) also appeared only 4 times. Each of the example instances we show occurred twice.

The second reason we used clustering was to obtain generalized patterns, which we can apply not only to the exact words which we found in the data, but to words similar to the ones we encountered. In all the examples mentioned above, the fact that the words belong to the same cluster allows us to detect a pattern that applies to many words in the cluster. For instance, in the first example stated above, many of the other verbs in the cluster (for example 'decrease') fit very naturally into the pattern, since they represent monetary actions which are likely to be taken by large companies. Even though we never observed these verbs as participating in the pattern in our corpus (it seems the companies described in the text tend to 'slash' their prices, but not 'decrease' them!), if we see them in some other text, we will recognize the pattern. This is true for all of the examples given above, and also for cases where some instances of the pattern *would* have been detected as significant, but no way existed to link these instances together, and see the unifying theme behind them.

5 Discussion

5.1 Conclusions

There are several conclusions we can draw from the results of our experiments. First, we conclude that combining clustering with structural information can result in better clusters, which capture a more generalized similarity that goes beyond the exact features used in the clustering. Second, such a combination can be used to successfully detect patterns at different levels of the data, and of varying complexity. Finally, we have seen that the clustering provides a means of detecting patterns in cases of data sparseness, and results in generalized patterns which allow us to handle instances that did not occur in the data using their similarity on the cluster level.

Additionally, in the area of natural language study, we have shown some interesting phenomena, regarding the way clauses are built, and how sentences in sequence are connected to one another. We have also shown that the clustering and model we chose for the purpose was successful in detecting these phenomena, and may be of use in other aspects of natural language research.

5.2 Other Areas of Application

Despite the strong focus we put on natural language tasks in this work, the framework presented here is applicable for other types of data as well. There are many areas of research which are devoted to searching for patterns in data which has some sort of sequential aspect. In fact, pattern searching is employed in many complex systems which have cause-and-effect properties, but where the precise rules are not immediately obvious, or too complex to analyze or formulate in detail. Two obvious examples are gene expression data, and stock exchange data, which have a temporal aspect with strong connectivity between past, present and future. Using clustering (according to some feature which is deemed likely to be relevant to unit interactions) to reduce the number of variables we are dealing with, to strengthen weak signals by grouping them together, and to smooth out perturbations, should be very effective. If we then add simple generalized models which would help us define the patterns we are looking for, the pattern searching task becomes much easier and more likely to yield significant results.

5.3 Possible Extensions and Improvements

5.3.1 Re-inserting Removed Words

As mentioned in the first part of the procedure description, the preprocessing of the data included the removal of all clauses where one of the component words appeared in the corpus less than one hundred times, for the reasons stated there. This removal presents some difficulties to our task of pattern recognition, especially with regard to patterns spanning more than one sentence. It creates 'holes' in the data in the form of missing clauses, which reduces the number of pattern instances, and, more importantly, distorts the true distances between two clauses. The obvious solution to these problems is to re-insert these missing clauses. This requires that we assign each of the words we eliminated to an existing cluster, so that the clauses they participate in can be described by a legal value of a clustered clause. The assignment is quite easily done by applying the iteration step of the *sIB* algorithm to each word we wish to assign, i.e. finding the assignment that most increases (or least decreases) the value of the IB functional (see section 2.3.1). The only problem with this approach is that, with words having a low number of occurrences in the data, our confidence in the accuracy of the assignment is low.

5.3.2 Different Data Set

In this work we focused on patterns representing relationships between clustered clauses, which were designed to model a description of an action in the text. We were looking for patterns which may represent cause-and-effect relations, or mutual effects of a single action. Unfortunately, the corpus we used does not describe day-to-day actions which might better fit most people's definition of *world knowledge*. The actions described in the text are often of a large scale, performed by and involving corporations, countries, natural disasters or large groups of people. In addition, there were several parts of the corpus which were unsuitable for our task, and interfered with our results, specifically the financial reports, and the docking manifests. If we were interested in automated acquisition of more mundane and basic world knowledge, involving day to day actions, it would be a good idea to use a corpus composed of stories and descriptions of everyday life, such as is found in realistic fiction. Another advantage of such a corpus is that there is much more continuity than is found in a collection of short news articles, allowing us to search for longer-range patterns.

5.3.3 Different Evaluation Method

In order to evaluate the significance of the patterns we found, we used the statistical *p-value* measure. We chose this method since it is intuitive and relatively simple, and provides a formal statistical proof of the significance of our results, besides giving us a way of selecting and scoring the patterns. Our pattern searching task can be seen as a form of data mining, in which case the pattern evaluation method using *support* and *confidence* measures, which is frequently used in the field of data mining, may be more appropriate. This method also provides a measure of how likely we are to encounter the second part of the pattern assuming we've seen the first part, which is required for many types of calculations.

5.3.4 Richer Sentence Model

In this work, we have only touched the tip (or root) of the parse-tree for each sentence or clause. We used only the subject-verb-object triangle in our model, for the reasons mentioned in the introduction (section 1.4) and in section 2.1.2. As a further step along the path we began in this work, it would be interesting to gradually add to the clause model, and enrich the types of relationship patterns we look for.

For example, a logical next step would be to add adjective and adverb modifiers to the clause structure, and examine the relationships between the original subjects, verbs and objects and their modifiers. It is quite obvious that nouns can be classified by the properties represented by their modifiers, such as "heavy objects", "expensive objects" and "soft objects" (in fact, many people may find this a more natural classification). What may not be so obvious, and may prove interesting, is the relationship between, for instance, the verb and the modifiers of its object (preliminary evidence on this type of relationship can be found in section 4.1.2). For example, the verb 'to lift' will probably occur with objects which are 'heavy', 'large' or 'unwieldy'. The noun 'dentist' may occur frequently in the neighborhood of 'painful'. From this information it may be possible for a program to automatically "understand" that the possibility of lifting may be subject to the object's weight and size, and maybe even to "understand" such expressions as "moving mountains", or similes such as "like having your tooth pulled"!

6 Appendix A – Clustering results

6.1 Subject Clusters

Cluster 1: gathering, column, message, route, task, location, variety, music, stream, coverage

Cluster 2: fis, islamists, democracy, repair, hearing, proceeding, preparation, signing, ceremony, voting

Cluster 3: tse, ceasefire, ipe, bi, nasdaq, truce, blue-chips, notice, fix

Cluster 4: open, fifth, take, third, second, fourth, select

Cluster 5: walkout, closure, disruption, freeze, drought, stoppage, blockade, postponement, boycott, cancellation, fluctuation, publicity, scare, shutdown, outage, illness, strike

Cluster 6: basis, cutout, minn, raw, reference, lean

Cluster 7: attache, onic, nymex, bureau, igc, commerce, secretariat, usda, wong, schneider

Cluster 8: asian, philippine, indonesian, tokyo, n.y., thai, singapore, mauritius, malaysian, brazilian

Cluster 9: string, disclosure, defection, finding, accusation, testimony, word, affair, batch, enlargement, behaviour, allegation, criticism, scene, turnout, revelation

Cluster 10: willingness, reluctance, mass, want, opportunity, saying, intention, permission, either, and, refusal

Cluster 11: goods, fleet, property, gas, commodity, content, computer, resource, brand, portion, material, cable, equipment, account, chip, car, vehicle, product, detail, component, portfolio, value, class

Cluster 12: poll, forecast, audit, data, calculation, statistic, reading, research, indicator, chart, survey, table, study, figure, evidence, tankan, analysis, projection, indication, estimate, report, document

Cluster 13: m1, reuter, story, pay, registration, reuters

Cluster 14: buying, decline, trend, intervention, jump, strength, expectation, increase, return, dip, optimism, pace, surge, availability, slide, gain, drop, demand, weakness, improvement, growth, opening, upturn, bounce, recovery, correction, performance, boom, advance, news, resumption, start, pickup, falls, rally, arrival, rebound, turnaround, rise, expiry, inflow, injection

Cluster 15: bullet, joint, missile, rocket, shot, ball, type, warrant

Cluster 16: 1-0, apec, african, aggregate, asean, 2-0, distillates

Cluster 17: broker, analyst, meteorologist, banker, watcher, ceo, forecaster, commentator, respondent, expert, specialist, observer, historian, chartist, insider, trader, strategist, pollster, economist, oecd, dealer

Cluster 18: liffe, santos, ups, stockpile, port, world, airport, hub, traffic, terminal, cooperative, canadian, u.s., us

Cluster 19: deputy, neighbour, politician, liberal, coalition, parliamentarian, bjp, ally, conservative, opponent, unionist, socialist, candidate, republican, foe, helms, faction, spd, democrat, mp, rival, labour, party, perot, senator, communist, businessmen, anc, congressmen, opposition

Cluster 20: sport, english, sponsor, taxpayer, professional, successor, punter, culture, rugby

Cluster 21: jew, immigrant, iraqi, zairean, kurds, hutus, rwandan, refugee, villager, civilian, pilgrim, enemy

Cluster 22: weather, cold, delay, damage, rainfall, rains, snow, temperature, rain, shortfall, shower, frost, ice, spell, congestion

Cluster 23: athlete, houston, lion, brave, astronaut, horse, tyson, rider, bull, queensland, bear, miami

Cluster 24: cent, spot, fob, meal, bunker, refund

Cluster 25: covering, suncor, saw, percent, light, moderate, reporting

Cluster 26: tanker, ship, freight, tonnage, cargo, freighter, vessel, convoy, ferry

Cluster 27: critic, commentary, editorial, greenpeace, industrialist, cleric, amnesty, environmentalist, follower, advocate, lobby, campaigner

Cluster 28: first, ira, other, indian, jakarta, style, contribution

Cluster 29: tractebel, suez, unilever, fiat, solvay, vnu, telefonica, anglo, lvmh, granada, eaux, ing, pearson, ntt, bskyb, rwe, veba, olivetti, audi, stet, telekom, polygram, sab, gencor, endesa, bass, emi, havas, lonrho, hoechst, thyssen, axa

Cluster 30: priest, actor, journalist, suspect, man, resident, relative, father, child, mother, briton, boy, girl, nurse, wife, businessman, survivor, policeman, son, colleague, simpson, daughter, veteran, officer, husband, dissident, teenager, woman, brother, friend, person, sergeant

Cluster 31: 100, equivalent, bel-20, topix, baltic, ordinary, explorer, 50, aex

Cluster 32: rumor, swing, jitters, pressure, trouble, lack, controversy, turmoil, instability, rumour, downturn, tightness, woe, confusion, crisis, concern, row, shortage, dispute, unrest, speculation, scandal, tension, threat, problem, worry, uncertainty, fear, tightening, slowdown, doubt, absence

Cluster 33: nzsc, homer, nzse-40, double, nzse-10, single

Cluster 34: heart, custom, green, backer, kesri, abacha, buyoya, white, fine

Cluster 35: desire, effort, attempt, ability, promise, drive, helping, influence, ways, inability, hope, commitment, aim, determination, help, pledge, can

Cluster 36: kia, 3com, abb, dasa, lagardere, merck, siemens, columbia, bae, nynex, contractor, raytheon, alcatel, monsanto, unocal, shipyard, hughes, aerospatiale, kvaerner, barrick, thomson-csf

Cluster 37: havel, izetbegovic, holbrooke, jiang, demirel, pope, hwang, emir, stoyanov, barzani, mediator, kwasniewski, pontiff, hasina, sihanouk, maskhadov, tudjman, kennedy, suharto

Cluster 38: name, series, tie, hit, choice, time, goal, place, recommendation, status, campaign, appeal, life, chance, course, claim, role, request, application, push, title, century

Cluster 39: mercedes, automaker, honda, digital, carmaker, nissan, ibm, aol, hp, microsoft, xerox, dell, intel, apple, netscape, automakers, toyota, ford, motorola, gte, compaq, chrysler, sony, quaker, oracle, cisco, sears, fidelity, mazda, kellogg, sun, hewlett-packard, kodak, sprint, novell

Cluster 40: challenge, priority, cause, file, face, responsibility, language, label, objective

Cluster 41: sales, inflation, profit, wage, gdp, workforce, payroll, m3, unemployment, ppi, volume, bankruptcy, cpi, output, inventory, import, lending, consumption, turnover, export, salary, confidence, production

Cluster 42: requirement, curb, deadline, penalty, target, criteria, zone, mix, framework, quota, cap, date, ceiling, limit

Cluster 43: iliescu, amnuay, morris, jackson, ranariddh, hamilton, chan, grobbelaar, tang, plavsic, leghari, banharn, khatami, bhutto, widow, rao, costa

Cluster 44: borrowing, overhaul, assistance, funding, issuance, incentive, tax, aid, hedging, duty, redemption, subsidy, space, proceeds, maintenance, financing, guarantee, tariff, efficiency, relief

Cluster 45: florida, applicant, bidder, holder, creditor, stockholder, consortia, aborigine, bondholder, two-thirds, rome, nominee, shareholder

Cluster 46: collapse, leak, attack, outbreak, clash, raid, conflict, incident, killing, murder, violence, battle, shooting, epidemic, bombing, fighting, accident, massacre, war, debate, collision, disaster, crash

Cluster 47: air, highway, sea, bridge, formation, sky, road, tunnel, prison, independence

Cluster 48: devaluation, participation, use, merger, presence, membership, entry, adjustment, launch, combination, implementation, this, turn, hike, inclusion, revaluation, success, takeover, cut, purchase, development, euro, effect, introduction, competition, removal, consolidation, link, buildup, liberalisation, reduction, appreciation, change, depreciation, acquisition, conversion, expansion, split, addition, shift, deregulation

Cluster 49: administrator, turner, stewart, taylor, evans, moore, davies, williams, walsh, lord, murray, robinson, clark, abdullah, sullivan, campbell, morgan, thomas, li, matsushita, professor, witness, jones, chen, neill, wright, wang, davis, anwar, wilson, martin, vincent, smith, manager, anderson

Cluster 50: client, parent, household, million, subscriber, citizen, population, entrepreneur, finn, japanese, those, customer, pole, consumer, advertiser, shopper, individual, patient, couple, half, employee, user, family

Cluster 51: kohl, chavalit, netanyahu, minister, commander, coach, castro, chancellor, simitis, chirac, chretien, clinton, lebed, juppe, fernandez, mandela, general, tung, brown, sharif, kuchma, chubais, lawyer, chief, pataki, head, he, howard, gujral, cardoso, greenspan, arafat, attorney, menem, kabila, executive, blair, zedillo, speaker, albright, gingrich, fujimori, ramos, cook, gore, prodi, premier, mayor, yeltsin, baker, king, lukashenko, berisha, clarke, dole, zeroual, murdoch, erbakan, she, moi, chernomyrdin, commissioner, leader, governor, samper, reno, hashimoto, bucaram, president, mahathir, treasurer, lee, jospin, prince, kim

Cluster 52: lux, fft, fee, moody, par, int, lon, libor

Cluster 53: emu, dialogue, agreement, arrangement, integration, project, transaction, proposal, relationship, work, settlement, procedure, step, restructuring, system, budget, program, structure, reshuffle, offering, programme, policy, measure, reform, strategy, case, approach, package, initiative, transfer, promotion, scheme, deal, that, solution, cooperation, operation, extension, listing, plan, approval, exercise, reorganisation, regime, partnership, talk, process, phase

Cluster 54: mob, attacker, assailant, gunmen, gunman, settler, fundamentalist, gang, rioter, robber, terrorist, militant, eta, commando, bomber, militiamen, youth, guerrilla

Cluster 55: engineering, energy, chemical, generation, telecoms, retail, food, processing, telecom, telecommunication, mining, tobacco, banking, manufacturing, pharmaceutical

Cluster 56: benchmark, shrs, bolivar, stock, dlr, mos, indices, lumber, stocks, forint, share, jgbs, index, future, eurodollar, leu, peso, won, taka, rupee, yuan, real, lev, ftse, dax, escudo

Cluster 57: tour, des, wicket, leg, league, feeder, cricket, slaughter

Cluster 58: high, average, stg, ground, low, lead, potential, positive, cash, steady, lower, rising, eye, boost, rose

Cluster 59: tennessee, door, pm, record, adr, basket, subscription, imm

Cluster 60: soccer, form, outflow, surprise, god, advantage, hole

Cluster 61: residence, west, northeast, town, hours, headquarter, living, village, island, reporter, coast, camp, south, resort, north, east

Cluster 62: oth, tea, wool, petrol, iss, avg, card, maximum, lds

Cluster 63: quarter, agenda, alternative, list, reason, example, feature, captive, item, these, results, result, schedule, beginning, category, hostage

Cluster 64: 0830, digest-australian, 0800, 0930, 0645

Cluster 65: austrian, french, jordanian, egyptian, israeli, british, active, uk, shenzhen, russian, german, mexican, nigerian, czech, belgian, dutch, turk, swiss, european, manila, public, leading, hungarian

Cluster 66: peter, congressman, caller, bishop, editor, berlusconi, yilmaz, bossi, gonzalez, gaddafi, lafontaine, cavallo, branson, klaus, dehaene, patten, ciller, author, amato, lang

Cluster 67: body, department, u.n., watchdog, fda, brussels, court, nasd, regulator, fifa, agency, epa, judge, imf, justice, ftc, cftc, faa, ministry, office, un, irs, eu, authority, fcc, commission, supervisor, sec

Cluster 68: coup, bpd, jbri, sellers, period, jcr

Cluster 69: tonne, loader, bulk, protein, sulphur, liquid, nwe

Cluster 70: price, peseta, swap, rouble, zloty, shilling, differential, grade, fuel, gasoline, rand, rupiah, greenback, pound, guilder, dollar, dow, markka, yen, t-bonds, ringgit, mark, franc, shekel, bond, bunds, rate, crown, lira, naphtha, yield, baht, sterling, premium, brent, diffs, drachma, currency, gilt

Cluster 71: disagreement, principle, argument, suggestion, objection, achievement, idea, question, consideration, involvement, compensation, element

Cluster 72: weight, focus, bean, relation, debut, attention, cycle, april-march, marriage, everything, shopping

Cluster 73: brewery, chain, disney, shop, korea, hotel, distributor, pit, supermarket, store, casino, restaurant, entity, each, segment

Cluster 74: ltte, hardliner, separatist, hizbollah, exile, serbs, extremist, tutsis, hamas, nationalist, moslem, loyalist, palestinian, croats, christian, wing, chechens, arab

Cluster 75: visitor, defector, tourist, passenger, firefighters, cosmonaut, hunter, victim, guest, adult, traveller, crew, volunteer

Cluster 76: newcastle, middlesbrough, chelsea, milan, scotland, england, bayern, ajax, juventus, barcelona, ronaldo, yankee, monaco, bolivia, arsenal, psg, pri, wimbledon, liverpool, champion, natal, ranger

Cluster 77: coupon, bidding, maturity, t-bill, flat, jgb, 32, gross

Cluster 78: boutros-ghali, osce, hanoi, unhcr, association, embassy, icrc, teamsters, confederation, broadcaster, speaking, communique, organisers, controller, federation, kremlin, charity

Cluster 79: banco, indust, non-farm, ex, jornal

Cluster 80: mbia, fgic, fsa, native, ambac

Cluster 81: kerb, senior, pension, muni, planning, convertible

Cluster 82: ones, ecb, gulf, policy-makers, policymakers, fomc, studio, artist, minority, opec

Cluster 83: irish, australian, argentine, school, finnish, spanish, indicative, 10-yr, govt, italian, portuguese, lme

Cluster 84: finish, ballot, hold, shrink, particular

Cluster 85: ble, boe, insee, eurostat, istat, api, consob

Cluster 86: balance, deposit, saving, economy, activity, harvest, risk, asset, collection, impact, flow, order, liability, expenditure, receipt, crop, profitability, deficit, interest, income, productivity, reserve, debt, loan, credit, spending, investment, ratio, payment, revenue, cost, loss, capacity, benefit, shipment, sale, margin, supply, expense, surplus, exposure, earnings

Cluster 87: eurotunnel, amp, lufthansa, sabena, northwest, cathay, wal-mart, sas, qantas, dhl, airlines, sia, mcdonald's, aeroflot, twa, delta, valujet, swissair, virgin, fedex, carrier, klm, kmart, airline, ba

Cluster 88: nazi, lease, personnel, cartel, fate, gun, --, weapon, land

Cluster 89: river, wind, toll, corruption, worst, volcano, water, mountain

Cluster 90: techs, industrials, chicago, bolsa, forestry, counter, equity, frankfurt, kiwi, electronics, continent, helsinki, london, bourse, financials

Cluster 91: hog, und, common, amount, kuna, dwt, load, corporates

Cluster 92: concession, adrs, visa, permit, mandate, licence, license, patent, exemption

Cluster 93: colombia, zimbabwe, burma, zambia, panama, ghana, brazil, croatia, bulgaria, slovakia, poland, lithuania, thailand, peru, philippines, algeria, russia, turkey, yemen, kuwait, austria, nigeria, guatemala, kenya, vietnam, mongolia, estonia, angola, cambodia, egypt, india, norway, kazakhstan, latvia, chile, china, belarus, slovenia, albania, ukraine, cuba, jordan, canada, moldova, morocco, serbia, pakistan, yugoslavia, indonesia, cameroon, greece, ireland, tunisia, malaysia, hungary, lebanon, nepal, bangladesh, belgium, romania, taiwan, argentina, malta, nicaragua, oman, sweden, japan, venezuela, shanghai, ecuador, australia

Cluster 94: craft, truck, flight, warplanes, airliner, bus, submarine, aircraft, plane, train, boat, helicopter, jet, shuttle

Cluster 95: personal, minimum, international, forex, specialises, de

Cluster 96: summit, legislature, lawmaker, assembly, legislator, committee, subcommittee, chamber, delegate, duma, senate, cabinet, council, meeting, referendum, conference, panel, euro-mps, forum, parliament, congress, voter, board, house

Cluster 97: retreat, plunge, breach, burst, bout, spate, selling, squeeze, influx, slump, comments, reports, sell-off, profit-taking, spree, break, selloff, anticipation, enquiry, liquidation, stop, flurry

Cluster 98: underwriter, organization, cia, university, volkswagen, forwarders, viacom, nike, foundation, publisher, thomson, municipality, adb, hospital, mof

Cluster 99: storm, blast, quake, typhoon, cyclone, landslide, explosion, shelling, fire, hurricane, flood, blaze, bomb, earthquake, tremor

Cluster 100: negative, initial, dubai, reoffer, call, btp, bund

Cluster 101: old, weekly, oilseed, normal, 97, 96, 98, cereal

Cluster 102: steer, heifer, well, no, close, dbrs

Cluster 103: sentence, sanction, veto, embargo, restriction, injunction, burden, practice, surcharge, ban, nyse

Cluster 104: hoiupank, komercni, nafta, hansapank, rr, slovnaft, sharp, propane, podravka, unibanka, cez, vsz, pliva

Cluster 105: deep, flag, territory, colony, yellow, cloud

Cluster 106: dog, her, its, rescuer, his, diver, our, their

Cluster 107: iri, rbnz, bpa, pemex, transco, cwb, bookmaker, goverment, apv, halifax, rba, treasury, awb

Cluster 108: irna, interfax, pap, bernama, xinhua, times, antara, ina, hina, tass, pti, journal

Cluster 109: georgia, asia, street, finland, africa, borrower, romanian, national, bulgarian, iranian, chechnya

Cluster 110: plaintiff, letter, indictment, filing, lawsuit, motion, judgment, petition, suit, article, complaint

Cluster 111: magistrate, prosecution, detective, fbi, cbi, ec, bnb, tribunal, prosecutor

Cluster 112: percentage, lot, total, worth, csu, pct, monthly, parcel, little, block, barrel

Cluster 113: donor, tanzania, ottawa, labor, catholic, kiev, seoul, taipei, s.korea, boss, canberra, ankara, beijing, cyprus, uae, saudi, ruler

Cluster 114: peak, host, mid-south, sydney, presidency, stadium

Cluster 115: itc, selector, pentagon, nasa, referee, moment, uefa, commonwealth, wto, g7, uaw

Cluster 116: ten, supporter, fan, student, score, trucker, striker, crowd, miner, protester, activist, dozen, demonstrator, protestor, thousand, marcher, workers, teacher, worker, hundred, fishermen, driver, albanian

Cluster 117: jury, physician, connor, surgeon, hall, smoker, juror, lake, megawati, sister, writer, singer

Cluster 118: rtrs-australia, quebec, eib, cme, corp, deutsche, ebrd, nomura, dresdner, finmin, mexico, s.africa, portugal, ifc

Cluster 119: chaos, strain, virus, error, cigarette, disease, spill, exuberance, smoke, smoking

Cluster 120: signal, wall, beer, cell, dam, tree, gene, band, forest

Cluster 121: cargill, cpc, aramco, refco, rbi, adm, tupras, bp, ecopetrol, ccc, ioc, glencore, foreigner, tender, commercial

Cluster 122: hilton, citicorp, usair, ameritech, csx, loewen, itt, sumitomo, bre-x, ge, mercury, henkel, conrail, krupp, issuer, bnp, mci, lloyd's, ahmanson

Cluster 123: 8, institute, moody's, comex, ibca, 2p, standard

Cluster 124: farm, vendor, aspect, internet, outlet, human, baby, engine, pc, animal

Cluster 125: origin, medium, long, kingdom, small, short, cbot, heavy

Cluster 126: drilling, test, contact, inquiry, sample, probe, inspection, testing, search, check, negotiation, consultation, investigation, review, examination, discussion, trial

Cluster 127: employer, america, country, participant, society, europe, owner, nato, community, club, member, farmer, leadership, moscow, american, bosnia, administration, them, britain, paris, organisation, mission, negotiator, partner, star, majority, france, italy, israel, counterpart, staff, republic, union, side, washington, pilot, will, team, church, both, denmark, bonn, nation, state, management, germany, alliance, switzerland, government, grower, bloc, they, spain, major

Cluster 128: gluten, okla, phil, kansas, compound, nebraska, plain

Cluster 129: eurobond, overall, original, tranches, debenture, size, note, dividend, tranche, certificate

Cluster 130: ldp, left, s.korean, nld, welfare, centre-right, udf, home, zajedno, ppp

Cluster 131: saddam, mobutu, lady, designer, queen, dostum, milosevic, masood, him, founder, unita, deng, diana, himself

Cluster 132: fully, trimming, carcass, debt-indian, beef

Cluster 133: circumstance, sentiment, technicals, caution, condition, pricing, environment, climate, outlook, fundamental, attitude, emergence, liquidity, possibility, view, situation, difficulty, perception, sign, stability, tone, mood, prospect, factor, politics, fact

Cluster 134: hasan, bos, brien, soros, lewis, allen, phillips, reynolds, harris, hanson, bell, young, staples, foster

Cluster 135: 16, three, five, seven, 15, nine, 18, four, 30, eight, two, six, 11, 10, 12, 14, 13

Cluster 136: upgrade, fixing, confirmation, completion, assessment, rating, valuation, revision, decrease, response, placement, allocation, reaction, downgrade, count

Cluster 137: macedonia, hk, fish, toronto, madrid, warehouse, chg

Cluster 138: act, provision, guideline, declaration, bill, charter, code, constitution, legislation, decree, amendment, treaty, directive, text, resolution, clause, formula, regulation, draft, protocol, accord, pact, law, compromise, rule, convention

Cluster 139: option, meanwhile, while, much, range, position, holding, end, offer, delivery, level, more, issue, another, capital, term, sector, bid, rest, number, show, centre, power, trading, some, market, point, money, need, part, oil, session, support, forward, area, last, itself, movement, trade, all, due, one, security, contract, stake, most, job, set

Cluster 140: m2, interim, q1, net, yr, shr, q4, q3, q2, feedlot

Cluster 141: grouping, sum, matter, violation, layoff, abuse, crime, fraud, stage, charge, latter

Cluster 142: privatization, passage, invitation, surgery, nomination, reelection, acceptance, address, selection, conclusion, appearance, establishment, presentation

Cluster 143: barrier, planting, border, resistance, injury, holiday, fight, heat, remains, run, expiration, stay, momentum

Cluster 144: samsung, shell, mobil, placer, tosco, enel, mitsubishi, universal, enron, nec, pdvsa, texaco, petronas, hyundai, inco, daewoo, amoco, lukoil, saga, arco, continental, iberia, dupont, chevron, elf, petrobras, exxon, statoil, pertamina

Cluster 145: lb, jail, sight, 20, kg, euroyen

Cluster 146: limited, 000s, mge, red, nbm

Cluster 147: tribe, kdp, prisoner, inmate, tribesmen, kidnapper, hijacker, puk, mrta

Cluster 148: information, book, broadcast, publication, satellite, telephone, phone, advertising, room, grant, profile, page

Cluster 149: curve, gap, showing, popularity, volatility, discount, draw, spread, switch, arbitrage, backwardation, widening, reversal

Cluster 150: enough, ring, intelligence, builder, franchise, rig, bulldozer

Cluster 151: chang, captain, sampras, fox, norman, black, thompson, graf, nun, aleman, park, martinez, kelly, johnson, hill, hingis, seles, carey, wood, becker

Cluster 152: current, same, special, u.k., main, annual, key, final, private, top, domestic

Cluster 153: finance, health, access, tv, construction, safety, exploration, distribution, marketing, communication, shipping, tourism, building, insurance, defence, control, infrastructure, transport

Cluster 154: gec, westinghouse, telkom, brewer, nationsbank, abc, gan, reliance, bbc, nbc, cbs, safeway, rank, heineken, skoda, ncb

Cluster 155: press, magazine, interview, radio, paper, television, daily, liberation, post, statement, newspaper, voice, independent, media

Cluster 156: bnr, cnb, buba, fed, nbh, riksbank, nbs, c.bank, bundesbank, cenbank, boj, nbp, snb

Cluster 157: coal, 4, 3, steel, 2, 1, loading, cement, container, fertiliser

Cluster 158: bruton, cimoszewicz, rato, rubin, bolger, spokesmen, source, rexrodt, moussa, mccurry, spokeswoman, pas, santer, colonel, trichet, fischer, issing, perry, mitsuzuka, silguy, familiar, spokesman, chairman, aide, secretary, horn, chidambaram, peters, stals, george, waigel, ciampi, dini, juncker, fischler, director, burns, undersecretary, lott, official, persson, hundt, broek, arthuis, barshefsky, yastrzhembsky, summers, goh, diplomat, downer, tietmeyer, sakakibara, strauss-kahn, welteke, glickman, costello, kinkel, spokesperson

Cluster 159: ericsson, handelsbanken, telmex, volvo, zeneca, ipc, nokia, telebras, pldt, ues, electrolux, merita, s-e-banken, guinness, nordbanken, astra, televisa

Cluster 160: treatment, drug, method, tactic, version, technology, software, design, device, vaccine, ending, tool, mechanism, technique, instrument, therapy, concept, model

Cluster 161: game, race, standing, play, cup, match, tournament, hand, championship, dealings

Cluster 162: buyer, investor, merchant, miller, mill, banks, importer, packer, refiner, lender, smelters, insurer, chinese, operator, institution, processor, player, speculator, wholesaler, issuers, roaster, seller, crusher, exporter, fund, shipper, local

Cluster 163: flotation, execution, core, western, ipo, float, auction, replacement, stable, pool

Cluster 164: bit, sound, rush, nothing, thing, thought, look, something, sort, anything

Cluster 165: pen, 408-8750, 31-20-504-5000, newsroom, 841-8938

Cluster 166: feed, alberta, central, belt, olein, distillate, prairie, saskatchewan, manitoba, correspondent

Cluster 167: troop, marine, military, bodyguard, peacekeeper, navy, slorc, guard, fighter, police, soldier, rebel, squad, taleban, patrol, force, militia, army, tank, policemen

Cluster 168: invite, il, roasting, deliverable, prefix

Cluster 169: fever, march, exodus, offensive, revolution, conviction, assault, protest, rebellion, confrontation, revolt, wave, riot, demonstration

Cluster 170: video, opinion, history, orders, picture, footage, breakdown, tape, screen, image, photograph, experience

Cluster 171: audience, importance, eps, consensus, capitalisation, beat, whole

Cluster 172: tiger, film, stand, front, inning, seed, seat, empire, pair, movie, swede, career, spaniard, favourite, winner

Cluster 173: effective, ch, bln, mln, max

Cluster 174: allianz, a.s., coca-cola, nestle, ici, b.a.t, bmw, raisio, ote, pioneer, westpac, plc, anz, mol, tesco, roche, novartis, philips, daimler, repsol, chase, bayer, ahold, renault, sap, creditanstalt, barclays, hsbc, basf

Cluster 175: bubor, min, buy, repurchase, former, repo, interbank

Cluster 176: quote, timing, speed, pattern, difference, award, prediction, nature, progress, scenario, continuation, extent, kind, easing, scale

Cluster 177: release, handling, obligation, quality, present, ownership, freedom, repayment, agriculture, purpose

Cluster 178: housing, pipe, mon, mortgage, base, employment, storage, fin

Cluster 179: 500, nikkei, 225, sub, 300, moving

Cluster 180: soy, maize, merch, texas, grain, diesel, pork, meat, primary

Cluster 181: track, feet, bag, cover, quantity, rail, destination, camera

Cluster 182: doctor, agent, engineer, consultant, nrc, inspector, scientist, auditor, advisor, investigator, technician, counsel, monitor, researcher, comptroller

Cluster 183: following, gainer, decliner, loser, dutroux, advancer

Cluster 184: bank, bt, bhp, developer, affiliate, provider, arm, branch, province, group, handful, conglomerate, retailer, region, utility, boeing, airbus, firm, manufacturer, carmakers, business, company, supplier, city, competitor, exchange, county, unit, trust, subsidiary, telstra, monopoly, syndicate, division, giant, maker, service, industry, gm, it, enterprise, producer, corporation, consortium, right, gazprom

Cluster 185: we, you, people, everybody, i, bowler, anybody, many, neither, me, everyone, viewer, none, others, any, kid, noone, few, someone, nobody, anyone, somebody, guy

Cluster 186: bryant, huang, tycoon, killer, hamanaka, defendant, thief, criminal, mcveigh, lopez, baron, accused

Cluster 187: luxembourg, fdp, tshisekedi, armenia, athens, netherlands, victoria, arnault, themselves, predecessor, dane, cdu

Cluster 188: channel, pipeline, factory, railway, center, site, smelter, network, generator, facility, platform, section, piece, reactor, line, complex, plant, station, field, refinery, venture, mine, machine

Cluster 189: santander, lehman, sbc, brokerage, care, salomon, paribas, ubs, cbank, merrill, citibank, csfb, bzw, rabobank, natwest, straight, commerzbank, painewebber, goldman

Cluster 190: peace, training, protection, cutting, buyback, creation, renewal, austerity, privatisation, education, disposal, convergence, transition

Cluster 191: csce, feedlots, michigan, informix, bse, corp., compuserve, eia, inc, inc., headline, fixture

Cluster 192: greek, slovenian, lithuanian, nz, delhi, polish, kl, swedish, chilean, turkish, bombay, danish, norwegian, bonus, per, preference, solidere

Cluster 193: bullion, copper, palladium, platinum, cotton, natgas, rice, canola, electricity, corn, rubber, soyoil, aluminium, crude, metal, coffee, wheat, zinc, back, tin, gold, barley, cattle, cocoa, sugar, soybean, nickel, silver

Cluster 194: edge, direction, upside, means, make, mkt, gmt, downside

Cluster 195: uganda, bahrain, rwanda, tehran, baghdad, vw, eritrea, zaire, opel, qatar, pyongyang, vatican, plo, sudan, libya, islamabad, dassault, ethiopia, iran, kinshasa, syria, iraq

Cluster 196: depos, floor, barge, cst, non-eu, cross, pellet, rotterdam

Cluster 197: minnesota, eastern, ore, california, illinois, district, midwest, ohio

Cluster 198: peres, brittan, camdessus, manuel, weizman, annan, nemtsov, fino, mbeki, aznar, cohen, richardson, rifkind, klima, ambassador, solana, aziz, coordinator, primakov, ikeda, secretary-general, levy, rodionov, hariri, delegation, christopher, kabariti, mubarak, adams, constantinescu, adviser, leary, vranitzky, al-sabah, ross, representative, envoy, rafsanjani

Cluster 199: decision, dismissal, warning, trip, arrest, announcement, death, stance, speech, defeat, setback, rejection, event, withdrawal, remark, vote, handover, victory, ruling, win, round, election, discovery, comment, action, move, suspension, failure, visit, verdict, outcome, resignation, appointment, crackdown, struggle, departure

Cluster 200: ad, advertisement, cd, window, parade, anniversary, exhibition, poster, festival

6.2 Verb Clusters

Cluster 1: intercept, rout, ambush, station, fire, dispatch, ally, oust, escort, humiliate, topple, disarm, deploy, overthrow, spray

Cluster 2: measure, track, float, debut, denominate, outperform, overvalue, undervalue, stable, shadow, capitalise, deplete, value, depreciate, class, lag, devalue

Cluster 3: relieve, extinguish, fan, cast, alleviate, override, soothe, quell, calm, dispel, allay, defuse, falter

Cluster 4: broker, seal, arrange, finalize, forge, clinch, reach, sign, renegotiate, negotiate, ditch, terminate, conclude, honour, strike, ratify, structure

Cluster 5: retake, seize, shell, liberate, recapture, overrun, attack, bombard, annex, pound, invade, bomb, capture, control, occupy, conquer

Cluster 6: burn, smash, burst, hurl, throw, blow, toss, explode, detonate, shatter, rip, destroy, tear, gut, wreck, rag

Cluster 7: age, drown, chant, dye, cry, brave, sex, proposition, don, wear, sing, dress

Cluster 8: save, reap, underwrite, lend, spend, redeem, owe, borrow, inject, drain, earmark, contribute, derive, invest, earn, accumulate, convert, refinance, repay, plough, allot, allocate

Cluster 9: drink, devote, eat, dedicate, bowl, sacrifice, overdo, breathe, smoke, bat, smell, swallow, live, talk, waste

Cluster 10: hoist, picture, overtake, hammer, thrash, crush, defeat, ride, trail, stick, wave, beat, lean

Cluster 11: rescind, seek, freeze, collect, win, withdraw, issue, deserve, accept, secure, revoke, grant, deliver, refuse, demand, award, distribute, obtain, solicit, receive, request, renew, withhold

Cluster 12: deflect, sidestep, poll, dodge, man, answer, skirt, beg, pitch, erect, survey, pinpoint, dismantle

Cluster 13: sport, space, cable, drug, shop, water, rail, pulp, franchise, page, pension, bus, phase

Cluster 14: trumpet, celebrate, score, attempt, decry, pronounce, plot, relish, decree, renounce, champion, mastermind, proclaim, commemorate, recount

Cluster 15: deplore, oppose, favor, favour, blame, criticize, condemn, defend, support, urge, blast, slam, declare, back, promise, welcome, laud, advocate, criticise, praise, abandon, hail, fight, tolerate, order, challenge, object, applaud, denounce, embrace, vow, protest, brand

Cluster 16: calculate, decrease, triple, revalue, inflate, lower, slash, trim, adjust, reduce, increase, halve, raise, hike, dilute, shave, swell, revise, double, pay, cut, generate

Cluster 17: schedule, reschedule, boycott, chair, reopen, attend, skip, joint, host, adjourn, inaugurate, convene, sponsor, pre-empt

Cluster 18: grade, clock, lade, cycle, sulphur, wet, nurse, date, crack, crane, fob

Cluster 19: taint, headline, chip, sun, key, radio, bite, band, size, traffic, straddle

Cluster 20: unload, park, hoard, dump, confiscate, pour, rent, retrieve, offload, divert, steal, insure, smuggle, subscribe

Cluster 21: release, register, book, incur, audit, post, record, reveal, tally, show, log, publish, detail, report, compile, restate

Cluster 22: place, use, base, remain, become, serve, offer, apply, switch, put, give, return, lose, lack, leave, work, change, turn, keep, make, pick, move, take, enjoy, like, choose, maintain, exchange, send, remove, run, have, replace, provide, target, establish, need, shift, set

Cluster 23: spawn, cause, provoke, stir, fuel, spur, stoke, arouse, ignite, trigger, induce, prompt, reignite, heighten, unleash, spark, lessen, revive, rekindle, inspire

Cluster 24: craft, ready, frame, row, snub, ram, lambaste, court, axe, bend, assail, confer, fault

Cluster 25: suppress, check, verify, engineer, invent, combat, practise, practice, clone, resent, condone, master

Cluster 26: zone, price, lot, spread, group, par, weight, average, rate, bale, yield, comprise, index

Cluster 27: overstate, surpass, eclipse, outdo, cushion, outstrip, offset, outnumber, equal, couple, overwhelm, exaggerate, exceed, outpace, brake, translate, outweigh

Cluster 28: refer, dismiss, act, press, investigate, dispute, refute, confirm, respond, plead, rest, deny, reject, sanction, stake, brush, probe, counter, ridicule

Cluster 29: syndicate, fish, attain, overshoot, better, satisfy, port, bypass, tariff, type, fulfil, fulfill, meet

Cluster 30: possess, delegate, reassert, misuse, relinquish, forfeit, squander, surrender, cede, retain, wield, invoke, exploit, abuse, exercise, amass, overstep, assert

Cluster 31: farm, implicate, indict, acquit, convict, jail, sentence, number, interview, exile, charge, fine

Cluster 32: pile, reassess, unwind, bear, exhaust, sever, square, reestablish, shorten, roll, leverage, exert, shoulder, liquidate, exit, hedge

Cluster 33: brew, center, surface, encompass, filter, accord, circulate, pop, correspond, centre, swirl, infect

Cluster 34: e-mail, bin, reference, denote, prod, except, wed, heat, prefix, tabulate

Cluster 35: transform, brace, behave, position, distinguish, align, busy, chain, rename, confine, pride, orient, rid, isolate, gear, distance

Cluster 36: lash, shake, blanket, cripple, engulf, paralyse, rock, grip, wash, rattle, ravage, jolt, flood, devastate, batter, swamp, sweep

Cluster 37: zero, melt, parallel, light, wipe, pattern, snap, moderate, segment, tail, barrel

Cluster 38: staff, boast, case, rival, trip, route, seat, slate, rank, qualify, shortlist, title

Cluster 39: upgrade, affirm, reaffirm, convey, downplay, assign, flag, misrepresent, sound, echo, reiterate, downgrade, repeat, relay

Cluster 40: kill, wound, injure, arrest, disperse, torture, detain, hunt, massacre, trap, kidnap, murder, abduct, slaughter, displace, slay, assault, shoot

Cluster 41: focus, undergo, concentrate, complete, simplify, pioneer, refocus, institute, speed, obstruct, wind, discontinue, supplement, undertake, relaunch, commence, coordinate, conduct, streamline

Cluster 42: ponder, absent, interpret, cheer, watch, bet, digest, ignore, disregard, discount, observe, await, eye, factor, wait, saw, fret

Cluster 43: film, punch, marry, lock, stab, hug, lay, dig, hole, hang, confess, bury, wreak

Cluster 44: coach, club, tour, depart, neighbour, cup, frequent, border, league, polish, visit, police

Cluster 45: contain, allow, bring, promote, further, link, represent, encourage, mean, enable, constitute, permit, force, entail, preclude, create, complement, cover, protect, ensure, aim, envisage, include, cost, guarantee, benefit, exclude, eliminate, prevent, follow, limit, require, involve, justify, prove, concern, combine

Cluster 46: congratulate, accuse, summon, invite, chide, appoint, name, instruct, notify, brief, advise, reelect, consult, elect, contact, inform, nominate, telephone, lobby, sue, tell, empower, ask, alert, sack, thank

Cluster 47: shed, recoup, reclaim, pare, extend, reverse, regain, pace, chalk, gain, claw, recover, consolidate, buck, retrace, erase, slice, notch

Cluster 48: hurt, depress, pressure, weaken, bolster, affect, squeeze, compare, drive, impact, buoy, restrain, drag, lift, upset, underpin, influence, knock, help, undercut, aid, propel, pull, boost, push

Cluster 49: specialize, overhaul, reform, install, advertise, incorporate, manufacture, package, integrate, access, exhibit, substitute, design, revamp, develop, license, patent, model

Cluster 50: okay, outlaw, authorize, regulate, exempt, privatize, oversee, bar, authorise, subsidise, supervise, deregulate, prohibit, forbid, liberalise, clear, vet, ban

Cluster 51: depict, expose, subject, shield, strangle, comfort, haul, disturb, thrust, burden, nett, sway, dwarf

Cluster 52: storm, search, ring, patrol, raid, loot, parade, comb, demolish, guard, scour, besiege, infiltrate, litter, surround

Cluster 53: grow, retreat, plunge, slip, steady, edge, rally, jump, sink, spike, total, dip, trade, rebind, slide, firm, drop, ease, rise, tumble, climb, stand, fall, sag, fell

Cluster 54: bank, retail, stock, bond, feed, labour, bill, grind, credit, state, power, paper, market, spot, share, barge

Cluster 55: classify, rush, commit, suspect, found, inspect, discover, unearth, clean, walk, uncover, hide, find, sort, count, identify, document, sit

Cluster 56: metal, oil, copper, gas, bunker, tin, grain, freight, rubber, sugar, wholesale, nickel, palm

Cluster 57: cross, flatten, breach, widen, hover, approach, near, touch, test, penetrate, narrow, revisit, break, pierce, carve, stabilize, bounce, hit, stay, scale

Cluster 58: channel, programme, finance, attract, draw, interest, relate, scheme, associate, dry, pool, soak, fund, absorb, appropriate, boom

Cluster 59: dock, sail, pilot, flight, fly, collide, ground, charter, hijack, ferry, circle, board, land, crash

Cluster 60: effect, jeopardize, constrain, inhibit, stimulate, restrict, disrupt, curb, hamper, curtail, stifle, necessitate, stem, hasten, facilitate, foreshadow, impede, accelerate, hinder, foster, distort

Cluster 61: treat, salute, neglect, hate, entertain, portray, betray, let, love, wish, commend, hand, rap, dislike, greet, succeed, question, confront, admire, trust

Cluster 62: confuse, deter, entice, disappoint, haunt, frighten, compensate, scare, worry, lure, discourage, catch, tempt, mislead, unnerve, unsettle, surprise

Cluster 63: safeguard, preserve, sour, brighten, threaten, harm, strengthen, strain, worsen, spoil, endanger, cloud, damage, impair, improve, enhance, tarnish, compromise, restore, undermine, ruin

Cluster 64: reorganise, build, own, list, diversify, broaden, merge, transfer, rebuild, shut, lease, reorganize, enlarge, acquire, sell, split, operate, expand, separate, restructure, purchase, divest, buy

Cluster 65: net, swap, provision, deposit, auction, shrink, reserve, gross, fetch, bid, loan, contract, tax, tender, refund

Cluster 66: tighten, relax, abolish, administer, clamp, evade, loosen, prescribe, levy, enforce, impose, waive, slap, stipulate, toughen

Cluster 67: store, handle, plant, export, refine, harvest, ship, consume, stockpile, donate, carry, pump, mine, import, produce, process, supply, transport

Cluster 68: truck, coal, berth, wire, bulk, iron, empty, bag, cement, load, anchor, discharge, steel

Cluster 69: overshadow, collapse, culminate, dog, pave, occur, accompany, fail, usher, plague, evolve, derail, precede, govern, mar, stall, span, complicate

Cluster 70: mull, study, envision, reconsider, analyse, assess, inherit, pursue, disclose, clarify, contemplate, analyze, discuss, examine, deem, recommend, decide, manage, specify, prefer, review, weigh, monitor, determine, evaluate, define, intend, explore, consider, assume

Cluster 71: contravene, adopt, modify, repeal, amend, pass, unveil, announce, approve, enact, propose, prepare, draft, endorse, devise, outline, submit, scrap, shelve, map, debate, vote, implement, present, introduce, violate, formulate, table, veto, reintroduce

Cluster 72: equip, connect, repair, locate, venture, network, service, assemble, accommodate, retire, designate, construct, house, part

Cluster 73: balance, espouse, adapt, copy, harden, resemble, soften, chart, fashion, rethink, tread, alter, dictate

Cluster 74: advance, tick, mix, expire, swing, end, open, tend, close, nudge, fix, settle, race, spin, roar, finish, skid

Cluster 75: spearhead, foil, intensify, abort, escalate, avert, doom, quash, prolong, thwart, wag, mount, repulse

Cluster 76: continue, begin, start, resume, hold, stage, cease, initiate, launch, halt, organise, delay, participate, cancel, block, kick, postpone, interrupt, stop, restart, plan, reinstate, suspend

Cluster 77: figure, season, people, second, last, level, black, round, match, deal, crown, feature, tie, top, even, correct

Cluster 78: colour, unify, contrast, merit, contradict, solidify, shape, belie, vindicate, infringe, validate, warrant, disguise

Cluster 79: regret, desire, afford, recognise, understand, perceive, view, achieve, learn, notice, play, mind, feel, accomplish, guess, regard, miss, hear, believe, try, forget, know, read, do, imagine, respect, judge, want, doubt, quantify, get, appreciate, can, remember

Cluster 80: bless, remind, prosecute, reassure, cheat, persuade, woo, convince, reimburse, impress, pardon, assure, insult, delight, punish, desert, teach, forgive

Cluster 81: activate, overbuy, reformulate, blend, minute, sharpen, sow, vary, short, oversell, sample, bottom

Cluster 82: diminish, dent, dampen, erode, sap, cool, temper, dull, thin, dash, cap, evoke, subdue, dim, mute

Cluster 83: argue, acknowledge, write, describe, recall, insist, term, resign, warn, reply, allege, speak, call, admit, claim, dub, concede, liken, mention, label, rule, hop, apologise

Cluster 84: reshuffle, dominate, re-enter, rejoin, direct, join, shun, form, disband, enter, lead, quit, dissolve, head, tap, phone, select

Cluster 85: resolve, suffer, encounter, withstand, survive, witness, weather, tackle, overcome, risk, sustain, address, face, avoid, inflict, solve, experience, endure, pose, escape

Cluster 86: air, sweeten, lodge, quote, forward, broadcast, render, update, file, screen, licence, mail

Cluster 87: body, captain, seed, field, research, moot, star, beleaguer, corner, stamp, mortgage, program

Cluster 88: please, divide, outrage, deprive, shock, embarrass, infuriate, elude, irritate, rob, pit, unite, silence, anger, enrage, stun, alarm

Cluster 89: interconnect, pelt, pen, fast, bone, compose, stone, sole, certificate, fax, scramble

Cluster 90: line, march, pack, cruise, mass, gather, blockade, crowd, stream, choke, flee, jam, scatter, strand

Cluster 91: fabricate, leak, transmit, print, spill, insert, flash, sift, subordinate, paint, extract

Cluster 92: imply, signify, highlight, mirror, exacerbate, mark, signal, underscore, compound, precipitate, mask, illustrate, herald, reinforce, suggest, underline, aggravate, reflect, demonstrate, spell, indicate, deepen

Cluster 93: expel, chase, recruit, hire, execute, airlift, arm, train, repatriate, evacuate, employ, rescue, extradite, shelter, engage, deport, free

Cluster 94: plug, best, bode, fill, drill, perform, cook, salvage, bridge, cop, augur, stretch, fit

Cluster 95: note, cite, peg, bemoan, lament, say, caution, tip, characterize, forecast, expect, detect, project, anticipate, attribute, comment, estimate, fear, see, explain, point, add, predict, speculate, foresee, trace

Cluster 96: spurn, decline, defy, retract, defer, televise, appeal, overrule, obey, resist, abide, heed, uphold, overturn, bow, rebuff

Cluster 97: cash, hog, crop, condition, mill, yellow, content, certify, white, corn, silver, spring

Cluster 98: disqualify, strip, frustrate, reward, guide, trouble, assist, alienate, spare, offend, bind, bother, suit, excite, entitle

Cluster 99: harbour, underestimate, realise, overestimate, display, emphasise, pin, realize, sense, overlook, express, conceal, voice, emphasize, stress, attach, recognize, gauge, communicate

Cluster 100: enlist, steer, manipulate, grab, contest, rig, prime, command, heap, muster, omit, snatch, annul, garner

6.3 Object Clusters

Cluster 1: bt, ups, microsoft, vw, apec, intel, apple, bre-x, imf, govt, gov, bundesbank, renault, g7, economist, sec, ba

Cluster 2: root, swipe, heart, turn, cue, step, toll, shape, knock, breather, twist, place, battering, gamble, precedence, beating, look, fright, precaution, tally, oath, shine, bite, tumble, dive

Cluster 3: bosnia, homeland, zaire, kuwait, cambodia, territory, colony, yugoslavia, cyprus, hebron, politics, scene, chechnya, afghanistan

Cluster 4: retreat, trend, slide, gain, climb, bounce, recovery, advance, downtrend, run-up, spree, rally, rebound, comeback, uptrend

Cluster 5: caution, resilience, custom, restraint, goodwill, reuter, tendency, distillate, sign, footage, usda

Cluster 6: fleet, affiliate, brewery, pipeline, factory, branch, center, chain, site, mill, shop, network, hotel, facility, centre, supermarket, section, unit, hub, location, reactor, subsidiary, store, outlet, complex, plant, station, field, refinery, division, office, casino, restaurant, block, warehouse, mine

Cluster 7: southwest, west, northeast, province, northwest, region, jerusalem, leaving, city, island, district, area, taiwan, coast, south, resort, north, east

Cluster 8: collapse, defection, tragedy, bout, spate, influx, erosion, downturn, worst, repeat, dilution, reprisal, shortfall, phenomenon, spike, flood, fallout, downgrade

Cluster 9: web, plot, tranches, aggregate, album, batch, estate, tranche, studio, wireless

Cluster 10: favour, serve, footing, nerve, wicket, ground, concentration, majority, possession, steam, foothold, composure, patience, momentum

Cluster 11: freight, overtime, hours, passenger, overhead, following, mail, hour, insurance, load, waste, shuttle

Cluster 12: throat, repos, lending, reverse, par, single, repo, ice

Cluster 13: curve, differential, industrials, dow, index, canada, future, spread, helsinki, bunds, yield, brent, backwardation, treasury

Cluster 14: sovereignty, dominance, logic, validity, legitimacy, freedom, discipline, viability, referendum, wisdom, innocence, credential, integrity, independence

Cluster 15: caretaker, pack, frn, estb, fix, gainer, decliner, loser, advancer

Cluster 16: troop, message, signal, letter, reinforcement, flame, fax, condolence, email, bulldozer, tank

Cluster 17: soy, trln, unq, alum, lumber, int, nickel

Cluster 18: forth, deadline, example, sight, calendar, schedule, foot, frame, sail, fire, tone, precedent, date, timetable

Cluster 19: coup, walkout, march, retaliation, placings, sit-in, stoppage, protest, boycott, rebellion, parade, revolt, uprising, demonstration, strike

Cluster 20: arm, flag, passport, banner, pistol, cocaine, knife, gun, baby, haul, arsenal, weapon, rifle

Cluster 21: best, solid, hero, living, worse, birdie, concrete, safe, bone, love, liquid, bad, public, favourite, convertible, hell, enemy, escape

Cluster 22: bank, sales, sport, asia, railway, bse, sector, utility, firm, trading, company, floor, market, sell, buy, trader, telecoms, telecom, telecommunication, remains, mining, trade, banking, enterprise, doing, dealer, top, airline, domestic, segment

Cluster 23: winning, fifth, ore, third, leg, second, fourth, lap, 24, sixth, straight

Cluster 24: 817-62-67, no.2, premia, 817-6267, 31-20-504-5040, mideast

Cluster 25: nose, ring, neck, leaf, crack, window, back, purpose, circle, tear, ace

Cluster 26: rumor, argument, suggestion, accusation, background, idea, rumour, notion, assertion, case, claim, allegation, contention, report, charge, theory

Cluster 27: bullion, liffe, kerb, ipe, istanbul, chicago, dlr, nymex, cme, shade, flat, spot, arabica, nasdaq, steady, lower, session, touch, toronto, lme, bourse, interbank, cbot, dealings, colombian, dax, gmt

Cluster 28: tick, cent, kobo, bps, lb, year-on-year, centavo, pct, cts, 2p, litas, percent, pfennig, kroons, tolars, santimes, pts, lat, month-on-month, bcf, notch

Cluster 29: stg, minimum, ecu, fraction, kuna, penny, bln, bfr, mln, ecus, punt, ringgit, shekel, crown, maximum, reais

Cluster 30: underwriter, asylum, autonomy, citizenship, certification, amnesty, waiver, injunction, visa, bail, permit, clarification, extradition, licence, listing, license, immunity

Cluster 31: manifesto, ad, publication, edition, advertisement, bulletin, defense, rescue, immigration, article, label, poster

Cluster 32: disruption, chaos, poverty, drought, recession, conflict, turmoil, delay, instability, woe, confusion, crisis, shortage, rift, pain, suffering, turbulence, hardship, congestion

Cluster 33: journey, arrangement, trip, effort, attempt, revolution, pullout, handover, sweep, preparation, filing, start, play, repatriation, redeployment, landing, visit, push, transition, process

Cluster 34: english, playoff, landmark, results, semifinal, premier, final, conclusion, quarterfinal, champion

Cluster 35: nikkei, benchmark, bu, normal, chase, carcass, key, ex, ftse, summary

Cluster 36: storm, backlash, bloodshed, exodus, jitters, friction, controversy, sell-off, row, unrest, speculation, outcry, frenzy, scare, selloff, panic, wave, uproar, flurry, furore

Cluster 37: strip, constituency, one-third, km, thirds, diamond, guide, remainder, two-thirds, stretch, whole

Cluster 38: common, intelligence, institute, airbus, packer, compliance, interim, cooperative, former, hurricane, monitor

Cluster 39: residence, courtroom, premise, university, campus, school, embassy, apartment, depot, port, headquarter, airport, hall, park, home, villa, church, compound, palace, hospital, building, square, prison, stadium, house

Cluster 40: sentiment, supervision, morale, ability, popularity, competitiveness, standing, liquidity, profitability, transparency, image, survival, effectiveness, efficiency, credibility

Cluster 41: belgrade, brussels, moscow, completion, paris, athens, sydney, close, bonn, vancouver, rome, miami

Cluster 42: colombia, finland, brazil, croatia, bulgaria, slovakia, poland, lithuania, thailand, peru, philippines, russia, britain, africa, austria, nigeria, vietnam, estonia, egypt, france, india, norway, italy, netherlands, chile, china, slovenia, albania, ukraine, jordan, morocco, indonesia, ireland, malaysia, mexico, hungary, denmark, belgium, romania, argentina, germany, sweden, portugal, japan, venezuela, spain, australia

Cluster 43: triple, sum, deposit, saving, wealth, million, reward, windfall, total, equivalent, average, extra, initial, double, amount, none, money, cash, tonnage, primary, billion, benefit, job

Cluster 44: medan, ara, delhi, bound, midday, rotterdam, manila, nwe, pinch

Cluster 45: actor, blue, tune, item, slogan, song, story, and

Cluster 46: fishing, speaking, thinking, broadcasting, playing, flowing, deliberation, rolling, flying, sitting, career, campaigning, smoking, coverage, racing

Cluster 47: agreement, declaration, settlement, lease, charter, decree, treaty, memorandum, multi, deal, protocol, accord, pact, contract, convention

Cluster 48: agent, engineer, consultant, specialist, brokerage, inspector, mercenary, nomura, professor, contractor, counsel, adviser

Cluster 49: eyebrow, spectre, markka, petrol, question, conc, uah

Cluster 50: rhetoric, trick, method, occasion, tactic, name, word, phrase, accounting, means, proceeds, methodology, technique, style, guidance, language

Cluster 51: buying, happening, burst, covering, selling, pick, switching, slowing, heading, rush, sky, profit-taking, sun, 97, liquidation, rising, widening, moving

Cluster 52: repair, audit, evaluation, drilling, test, work, rehabilitation, procedure, training, survey, inspection, testing, study, maintenance, surgery, check, analysis, transformation, review, examination, therapy, experiment

Cluster 53: cruise, same, 4-mar-97, nis, bowler, forever, prev, chord

Cluster 54: plunge, slip, end, edge, jump, dip, surge, drift, slump, finish, soar, sharp, leap, falls, ease, split, inch, rose

Cluster 55: 16, 70, 15, 18, 17, 19, 200, 30, 40, 65, 21, 23, 11, 20, 10, 22, 60, 25, 12, 50, 14, 13

Cluster 56: finding, offer, calculation, testimony, undertaking, comparison, speech, promise, bid, assessment, submission, interview, prediction, invitation, recommendation, appeal, donation, address, call, plea, request, presentation, pledge

Cluster 57: snap, exile, original, consensus, witness, bp, deng, warrant

Cluster 58: ten, three, five, seven, nine, florida, score, american, four, texas, israeli, guard, california, fighter, african, miner, eight, two, six, dozen, soviet, albanian

Cluster 59: coal, wine, concentrate, tea, beer, tyre, milk, cigarette, meat, steel, semiconductor, motorcycle, fish, chicken, produce, alcohol

Cluster 60: agenda, column, list, conglomerate, ranking, indicator, chart, table, headline

Cluster 61: pro, negative, clock, heat, ear, positive, upbeat, professional, tide, sour, corner, tail

Cluster 62: ceo, candidacy, bidder, candidate, remedy, successor, squad, slate, nomination, replacement, nominee, resignation

Cluster 63: kohl, netanyahu, chirac, clinton, lebed, juppe, jiang, mobutu, mandela, tung, pope, chubais, milosevic, howard, arafat, kabila, blair, albright, fujimori, prodi, yeltsin, lukashenko, berisha, erbakan, mubarak, chernomyrdin, hashimoto, lee, kim

Cluster 64: percentage, data, oat, pea, debenture, note, seed

Cluster 65: havoc, foundations, foundation, horn, wreath, groundwork, siege

Cluster 66: channel, basis, household, housing, engineering, tv, joint, special, paper, telephone, front, communication, core, generation, retail, main, cable, auto, tobacco, pool, private, manufacturing

Cluster 67: artillery, shell, bullet, volley, missile, rocket, shot, mortar, ammunition

Cluster 68: circumstance, origin, alternative, subject, matter, aspect, timing, issue, feasibility, possibility, ways, cause, fate, extent, topic

Cluster 69: concession, decision, disclosure, confession, announcement, judgement, remark, admission, ruling, judgment, representation, discovery, comment, selection, apology, verdict, revelation

Cluster 70: ash, rock, flower, bottle, grenade, object, explosive, egg, bomb, stone, glass

Cluster 71: liberal, whom, conservative, socialist, republican, defendant, democrat, predecessor, lopez, dole, communist, bhutto, pri

Cluster 72: ltd, update, receipt, highlight, continent, kernel, fim

Cluster 73: grip, belt, watch, silence, rein, vigil, finger, profile, eye, lid

Cluster 74: delivery, vacation, another, booking, ride, holiday, wind, lying, hit, time, moment, pass, life, break, pause, couple, kick, miracle, past, run, length

Cluster 75: gathering, journalist, interviewer, audience, seminar, rtl, magazine, travelling, newsletter, truth, radio, interfax, television, abc, crowd, luncheon, bbc, delegate, briefing, regular, reporter, conference, newspaper, cnn, journal, opec, correspondent, reuters

Cluster 76: imprisonment, bankruptcy, fight, wait, default, batter, battle, emergency, epidemic, barrage, discrimination, war, rivalry, disaster, struggle

Cluster 77: ita, sep, fra, u.k., manuf, m4

Cluster 78: assistance, funding, term, raise, cover, pension, aid, protection, subsidy, allowance, rebate, allocation, grant, loan, one-off, credit, compensation, financing, subscription, payment, guarantee, pay, lev, stay, exemption, certificate

Cluster 79: upgrade, recall, transaction, retirement, series, cutback, inquiry, shake-up, exit, lineup, redemption, reshuffle, offering, ipo, switch, promotion, enquiry

Cluster 80: deputy, vice, press, plaintiff, u.n., algerian, navy, justice, spokesman, police, plo, times, attorney, ministry, saudi, iranian, militia, cuban, kremlin, media

Cluster 81: country, europe, who, community, club, department, watchdog, organization, leadership, group, coalition, labor, military, administration, regulator, association, organisation, agency, world, institution, others, faction, union, side, team, labour, giant, party, federation, authority, nation, state, management, corporation, government, army, wing, minority, family, major

Cluster 82: release, devaluation, participation, abolition, membership, integration, intervention, entry, launch, implementation, adoption, hike, inclusion, passage, withdrawal, opening, cut, setting, modification, formation, introduction, revision, signing, enlargement, breakup, removal, resumption, deployment, creation, renewal, reduction, transfer, change, lifting, move, restoration, extension, establishment, ratification, outcome, appointment, departure

Cluster 83: goods, grade, fuel, gas, ticket, worth, diesel, electricity, mortgage, barge, piece, submarine, cargo, metal, oil, bean, material, quantity, bushel, gold, parcel, pair, wood, barrel

Cluster 84: fever, leak, pollution, virus, flaw, brain, infection, outages, disease, fault, symptom, gene, illness, cancer

Cluster 85: burma, uganda, bahrain, rwanda, tehran, baghdad, turkey, burundi, israel, sudan, libya, cuba, washington, serbia, beijing, pakistan, other, greece, switzerland, iran, syria, iraq

Cluster 86: america, ally, star, anybody, republic, kid, defender, god, someone, king, young, somebody, friend, prince, guy

Cluster 87: rouble, zloty, rand, shrs, rupiah, stock, pound, dollar, share, jgbs, yen, peso, mark, franc, bond, rupee, basket, lira, yuan, baht, sterling, currency

Cluster 88: vacuum, carmaker, impasse, value-usda, mir, taxpayer, deadlock, gm, stalemate

Cluster 89: employer, you, supporter, fan, parent, doctor, analyst, everybody, her, resident, politician, them, me, everyone, viewer, colleague, romanian, mp, speculator, follower, him, anyone, businessmen, voter, us, shareholder

Cluster 90: utmost, okay, so, santos, enough, my, wrong, brasil, well, opposite, nothing, thing, what, our, stuff, little, something, sul, everything, doe, can, whatever, anything

Cluster 91: eurobond, warning, emtn, summons, communique, alert, denial, subpoena, eurobonds, statement, ultimatum

Cluster 92: decline, stabilisation, swing, increase, revival, acceleration, volatility, doubling, drop, deterioration, weakness, improvement, outflow, growth, upturn, correction, boom, decrease, consolidation, retracement, build, pickup, continuation, buildup, slight, fluctuation, appreciation, movement, softness, pullback, depreciation, turnaround, reaction, tightening, rise, slowdown, contraction, inflow, easing, shift, reversal

Cluster 93: track, pace, speed, pattern, path, separate, processing, patrol, bull, bear, trail, course, distance

Cluster 94: condition, requirement, obligation, target, precondition, criteria, goal, standard, norm, quota, specification, levy, maastricht, criterion, cap, ceiling, limit, objective

Cluster 95: ill, approx, prey, victim, short, shy, foul

Cluster 96: overhaul, privatization, merger, reconstruction, spinoff, project, reorganization, construction, buyout, spin-off, flotation, issuance, takeover, restructuring, purchase, program, reform, float, issuing, buyback, privatisations, repurchase, liberalisation, buy-back, placement, privatisation, acquisition, disposal, conversion, expansion, tie-up, reorganisation, deregulation

Cluster 97: herself, body, yourself, ourselves, busang, survivor, trace, simpson, myself, themselves, mcveigh, itself, himself

Cluster 98: appetite, enthusiasm, strength, clout, dream, influence, attraction, bias, potential, ambition, correlation, political, advantage, interest, regard, loyalty, sympathy, chance, hope, memory, faith, respect, confidence

Cluster 99: rev, dm, stocks, shr, n.a., div, nil

Cluster 100: newcastle, milan, england, juventus, newsroom, barcelona, desk, liverpool, ranger

Cluster 101: air, drug, battery, generator, pump, vaccine, sample, chemical, drink, display, food, protein, water, medicine, blood, oxygen

Cluster 102: osce, court, donor, jury, legislature, lawmaker, judge, assembly, legislator, committee, chamber, juror, senate, cabinet, council, panel, tribunal, investigator, parliament, congress, fcc, commission, prosecutor

Cluster 103: finance, farm, broadcast, energy, health, grain, research, commodity, aviation, allotment, affair, take, exchange, county, commerce, trust, western, forex, planning, agriculture, de, gulf, shipping, tourism, industry, one, defence, security, leading, independent, transport, local

Cluster 104: volume, layoff, stockpile, content, equal, population, inventory, liability, valuation, backlog, turnover, income, reserve, debt, storage, ratio, capitalisation, revenue, cost, proportion, capacity, margin, expense, exposure

Cluster 105: range-bound, quiet, mystery, neutral, sound, near-term, active, rangebound, seller, light, calm, stable, slack, moderate, performer

Cluster 106: film, video, opinion, copy, map, photo, card, picture, file, tape, screen, movie, photograph, page, document

Cluster 107: triumph, milestone, achievement, birthday, breakthrough, victory, beginning, win, try, anniversary, century

Cluster 108: outbreak, clash, blast, death, incident, killing, violence, shooting, explosion, destruction, scandal, bombing, fighting, accident, massacre, kidnapping, blaze, riot, crash

Cluster 109: islamists, suspect, separatist, immigrant, killer, serbs, businessman, activist, gang, comrade, criminal, dissident, terrorist, spy, militant, fishermen, mrta

Cluster 110: birth, hint, variation, go-ahead, fda, reason, blessing, better, welcome, nod, ok, hurt, indication, breakdown, lift, detail, boost, mkt

Cluster 111: craft, dog, brief, lesson, tip, cow, publicity, trademark, art, patent

Cluster 112: ing, iowa-so, stockholder, 1.1-for-1, ml, provs

Cluster 113: gap, book, mob, niche, loophole, rank, hair, arbitrage, band, horizon, silver

Cluster 114: maturing, indices, pln, wi, strategist, btp, gilt, bund

Cluster 115: counter-offensive, attack, raid, offensive, genocide, drive, counterattack, crusade, assault, blitz, search, campaign, hunt, occupation, crackdown

Cluster 116: friendship, presence, tradition, habit, relationship, contact, tie, reputation, monarchy, capability, relation, monopoly, privilege, link, status, spirit, luxury

Cluster 117: prize, medal, vote, reprieve, landslide, trophy, oscar, seat, toss, election, runoff, reelection, praise, honour, title

Cluster 118: 100, weight, corp, international, gear, fob, inc, electronics, rig, petroleum

Cluster 119: closure, dismissal, assassination, arrest, prosecution, halt, expulsion, capture, execution, postponement, seizure, deportation, cancellation, exclusion, detention, shutdown, evacuation, suspension

Cluster 120: hizbollah, zairean, kurds, tutsis, ira, rwandan, rebel, taleban, eta, moslem, kurdish, hutu, islamist, loyalist, christian, guerrilla

Cluster 121: option, property, frequency, holding, shareholding, capital, asset, franchise, power, resource, slice, ownership, portion, space, base, slot, fund, portfolio, right, stake, land, chunk

Cluster 122: clothing, hat, shirt, colour, toy, uniform, excess, shoe, white, cloth, dress

Cluster 123: abortion, sponsorship, handling, conduct, welfare, veto, human, behaviour, practice, marriage, refusal

Cluster 124: stabilise, relative, korean, offset, fortune, in, saw, present, cross, czech, make, middle, red, commercial, heavy

Cluster 125: quote, pitch, sense, distinction, error, difference, stride, mistake, headway, indicative, mention, secret, progress, sacrifice, debut, homeless, reference, noise, contribution, gesture, inroad, appearance

Cluster 126: cob, scoreboard, pellet, bulk, barley, cattle, feeder, slaughter

Cluster 127: pressure, strain, squeeze, spin, stress, stamp, damper, emphasis, blame, forward, dent, bet, burden, stop, brake

Cluster 128: insistence, medium, ibm, source, zero, nigerian, destination, anticipation, pulp, gateway, motor, textile

Cluster 129: sigh, mon, deviation, derivativesdesk, scorn, hide

Cluster 130: millfeeds, berth, discharger, discharge, terminal, meal, loader, fio, cement, fixture, flour, fertiliser

Cluster 131: first, quarter, showing, while, much, lot, range, ones, bit, current, more, its, this, rest, reading, show, some, business, many, point, overall, these, which, his, small, saying, those, any, news, no, thought, few, will, last, whose, account, both, all, due, it, either, their, face, real, half, most, each, sort, kind, set, fact

Cluster 132: machinery, programming, satellite, server, internet, technology, software, derivative, device, platform, entertainment, database, computer, modem, system, phone, hedge, brand, processor, tool, equipment, chip, service, instrument, engine, pc, product, variety, array, component, camera, type, hardware, machine, infrastructure, model

Cluster 133: existence, nationality, conspiracy, involvement, api, identity, wrongdoing

Cluster 134: desire, resolve, willingness, belief, reluctance, lack, rejection, conviction, solidarity, readiness, intent, commitment, wish, failure, determination, preference, absence

Cluster 135: semi-ann, coupon, rent, principal, annual, quarterly, freq, arrears, tribute, ransom

Cluster 136: emu, nato, grouping, chorus, erm, fray, cartel, asean, venture, force, wto, eu, consortium, alliance, bloc, grid, partnership

Cluster 137: palladium, fixing, platinum, natgas, t-bill, propane, dispute, auction, litigation, prompt, expiry

Cluster 138: price, wage, discount, tax, payout, duty, royalty, fee, fare, vat, dividend, salary, rate, taxation, premium, tariff, bonus, multiple, refund

Cluster 139: mth, mind, h1, q1, yr, landscape, q4, 96, q3, hand, q2

Cluster 140: function, concert, lunch, mass, dinner, ceremony, forum, prayer, fruit, funeral, reception, brunt

Cluster 141: feet, tonne, bpd, 20,000, 100,000, 1,000, 50,000, 10,000, bale, bag, 2,000, estimated, further, 40,000, kg

Cluster 142: tennis, game, soccer, host, football, golf, match, cricket, role, music, catch, rugby

Cluster 143: roof, tower, statue, grave, tree, pole, mountain, hole, debris, forest

Cluster 144: open, bidding, slalom, race, men's, contest, cup, league, tournament, championship, prefix, wimbledon

Cluster 145: we, senior, yes, statistic, spokeswoman, nikko, economics, pit, fed, he, livestock, syndicate, she, that, they, feedlot

Cluster 146: people, jew, student, man, father, mother, boy, girl, wife, policeman, son, protester, soldier, settler, daughter, husband, woman, demonstrator, villager, teacher, bomber, brother, civilian, palestinian, youth, person, policemen, arab

Cluster 147: pocket, weather, excitement, cold, upside, justification, synergy, rainfall, snow, fit, temperature, room, rain, shower, frost, smoke, moisture, stream, scope, downside

Cluster 148: guideline, principle, proposal, bill, outline, version, format, code, design, constitution, legislation, amendment, budget, framework, programme, directive, strategy, text, resolution, mini-budget, formula, package, regulation, draft, scheme, plan, law, blueprint, rule, concept

Cluster 149: nz, roll, entire, shoulder, want, mainland, medicare, accused, knee

Cluster 150: poll, tour, routine, hearing, summit, proceeding, roadshow, celebration, mission, event, probe, meeting, negotiation, consultation, round, investigation, exhibition, discussion, debate, trial, festival, tender, exercise, talk, voting

Cluster 151: balance, environment, return, economy, climate, combination, outlook, safety, fundamental, success, quality, performance, continuity, structure, mix, coordination, situation, scenario, atmosphere, prosperity, perception, stability, mood, prospect, convergence, culture, degree, value, scale

Cluster 152: mosque, town, stronghold, grozny, enclave, village, arbil, kabul, lebanon, kisangani, hill, camp, kinshasa, goma

Cluster 153: burn, wound, harm, embarrassment, defeat, setback, injury, damage, casualty, stroke, blow

Cluster 154: bribe, leave, shelter, bargain, revenge, gift, care, kickback, comfort, query, refuge, encouragement

Cluster 155: inflation, borrowing, usage, workforce, forecast, expectation, unemployment, availability, number, demand, output, throughput, expenditure, dependence, odds, likelihood, size, reliance, consumption, deficit, productivity, projection, estimate, spending, emission, supply

Cluster 156: waiting, 4, 3, 2, mineral, 1, loading, vessel, wharf, container

Cluster 157: prisoner, captive, inmate, kashmir, breath, post, presidency, command, hostage, responsibility

Cluster 158: treatment, signature, confirmation, expression, advice, award, simple, order, recognition, consideration, clearance, input, authorization, dose, consent, endorsement, response, authorisation, backing, permission, notice, reply, acceptance, notification, instruction, mandate, help, approval, assurance, injection

Cluster 159: steer, pig, heifer, sheep, adult, animal, class

Cluster 160: broker, client, buyer, developer, owner, provider, investor, farmer, importer, retailer, holder, distributor, manufacturer, partner, refiner, lender, supplier, insurer, operator, creditor, borrower, player, customer, issuers, consumer, maker, carrier, producer, exporter, user, grower

Cluster 161: society, haven, reality, venue, feature, priority, focus, mechanism, category, theme, description, entity, definition, factor, symbol, element

Cluster 162: freeze, curb, jail, sentence, penalty, sanction, blockade, invasion, embargo, restriction, punishment, moratorium, austerity, curfew, fine, surcharge, ban

Cluster 163: gloom, dem, shadow, ballot, cloud, chf, sfr

Cluster 164: participant, athlete, visitor, tourist, citizen, wounded, black, handful, personnel, child, observer, staff, scientist, veteran, national, pilot, refugee, thousand, guest, patient, employee, worker, foreigner, hundred, crew, driver

Cluster 165: captain, minister, commander, coach, chancellor, member, banker, general, expert, negotiator, striker, lawyer, chief, head, chairman, aide, counterpart, secretary, ambassador, executive, officer, director, speaker, official, mayor, delegation, senator, diplomat, leader, governor, president, manager, representative, envoy

Cluster 166: 3-0, 5-0, 1-0, 5-1, 4-1, 4-0, 2-0, 4-2, 3-2, 3-1, 2-1

Cluster 167: critic, neighbour, opponent, competitor, foe, horse, rival, nationalist, zajedno, lobby, opposition, winner, ruler

Cluster 168: pre-tax, profit, gdp, preliminary, eps, payroll, m3, oilseed, second-quarter, cpi, third-quarter, result, figure, net, draw, monthly, gross, loss, turnout, surplus, earnings

Cluster 169: provision, weekly, adjustment, installation, revaluation, assumption, qualification, separation, repayment, safeguard, clause, write-off, elimination, exception, addition

Cluster 170: impetus, impulse, information, explanation, specific, answer, access, certainty, incentive, clue, glimpse, reassurance, excuse, opportunity, backdrop, leeway, insight, flexibility, evidence, support, clarity, proof, leverage, stimulus, relief, respite, impression

Cluster 171: truck, tanker, airplane, route, ship, boeing, airliner, bus, aircraft, plane, freighter, train, car, boat, helicopter, vehicle, convoy, ferry, jet, van

Cluster 172: rating, a1, aaa, a2, athibid, bombay, cetes, aa, per

Cluster 173: greek, asian, philippine, irish, australian, austrian, argentine, french, finnish, egyptian, spanish, slovak, iraqi, indonesian, british, chinese, japanese, thai, russian, swedish, italian, german, mexican, turkish, malaysian, indian, belgian, canadian, bulgarian, portuguese, danish, dutch, u.s., swiss, european, pakistani, board, brazilian, hungarian

Cluster 174: bin, i, nic, olein, tic, publisher

Cluster 175: old, planting, harvest, acreage, uk, ending, crop, cutting, midwest, arrival, carryover, startup, shopping, cereal

Cluster 176: ericsson, brunei, eaux, stet, inc., du, qualifier, mci, thomson-csf

Cluster 177: reach, surface, central, thanks, duration, slows, fair, midnight, lie, commissioner

Cluster 178: module, harbour, lighthouse, smelter, cathode, dwt, empire, arena

Cluster 179: dialogue, ceasefire, peace, reunification, democracy, unity, mediation, reconciliation, euro, truce, confrontation, interpretation, solution, self-rule, cooperation, engagement, compromise, arbitration

Cluster 180: plug, passion, emotion, teeth, imagination, pride, punch, courage, muscle

Cluster 181: sea, green, putt, globe, belarus, daily, circuit, norwegian, mile, pilgrim

Cluster 182: checkpoint, barrier, puck, cordon, barricade, spotlight, tent, roadblock, limelight

Cluster 183: zimbabwe, bucharest, seoul, kenya, dhaka, tokyo, polish, hk, frankfurt, singapore, bangladesh, london, madrid, jakarta, warsaw, shanghai

Cluster 184: say, conversation, merit, history, trouble, maturity, impact, repercussion, weighting, neither, significance, meaning, choice, bb-minus, illusion, effect, intention, connection, feeling, plenty, bearing, feel, affect, consequence, sex, fun, jurisdiction, expiration, implication, experience, knowledge, motive

Cluster 185: regret, disagreement, disappointment, objection, optimism, scepticism, reservation, anxiety, frustration, dismay, surprise, alarm, concern, suspicion, satisfaction, outrage, shock, worry, dissatisfaction, fear, doubt, anger

Cluster 186: corridor, border, bridge, river, runway, frontier, cell, atlantic, zone, boundary, bar

Cluster 187: overs, deep, homer, wall, walk, counter, inning, ball, snag

Cluster 188: use, swap, flight, pricing, activity, collection, refining, flow, exploration, distribution, import, marketing, traffic, transportation, development, equity, advertising, employment, transmission, travel, export, rail, individual, education, investment, operation, production, shipment, sale

Cluster 189: position, form, stance, direction, lead, attitude, stand, part, line, policy, measure, view, hold, approach, initiative, action, thrust, aim, vow, control, regime

Cluster 190: vision, taste, understanding, subscriber, substance, luck, depth, dimension, character, characteristic, expertise, bed, perspective, skill, voice, ingredient, talent

Cluster 191: imbalance, awareness, importance, sensitivity, hostility, tightness, necessity, need, nervousness, nature, urgency, tension, uncertainty, probability

Cluster 192: highway, chapter, street, scoring, avenue, crossing, door, gate, entrance, box, road, tunnel, a.m.
Cluster 193: peak, threshold, high, level, low, intraday, parity, height, age, earth, stage, record, bottom, mph
Cluster 194: timber, pipe, sheet, no., wire, scrap, temptation, count
Cluster 195: wrath, 1-1, 0-0, parallel, attention, condemnation, scrutiny, criticism, 2-2
Cluster 196: hog, copper, maize, gasoline, feed, cotton, rice, wool, corn, rubber, pork, soyoil, aluminium, crude, belly, coffee, wheat, zinc, bunker, sulphur, cocoa, naphtha, sugar, soybean, beef
Cluster 197: tough, hurdle, risk, task, resistance, disadvantage, danger, challenge, obstacle, nightmare, headache, dilemma, competition, difficulty, threat, problem, limitation, complication, constraint
Cluster 198: censure, prospectus, indictment, shelf, lawsuit, motion, petition, suit, application, complaint, registration
Cluster 199: string, lull, standoff, era, period, collaboration, cycle, drama, streak, outage, spell, reporting, phase
Cluster 200: act, breach, violation, abuse, murder, corruption, terrorism, crime, incursion, fraud, torture, suicide, interference, aggression, irregularity, offence

7 Appendix B - Intra-Clause Patterns

For convenience, the acronyms SC, VC, and OC will be used when presenting the results to represent subject-cluster, verb-cluster, and object-cluster respectively. Therefore, the pattern "VC_23 – OC_12" means that (words from) Verb Cluster number 23 occurred with (words from) Object Cluster number 12.

7.1 Subject – Verb Patterns

```
SC_184 - VC_22  e.g.  "utility - need", "monopoly - set", "producer - use"
SC_185 - VC_22  e.g.  "others - put", "everybody - make", "anybody - base"
SC_127 - VC_22  e.g.  "leadership - remain", "moscow - use", "france - switch"
SC_51  - VC_22  e.g.  "murdoch - run", "yeltsin - set", "chancellor - set"
SC_51  - VC_46  e.g.  "general - inform", "executive - advise", "moi - thank"
SC_185 - VC_79  e.g.  "someone - read", "we - can", "anyone - imagine"
SC_17  - VC_95  e.g.  "trader - characterize", "forecaster - say", "broker - foresee"
SC_185 - VC_95  e.g.  "i - estimate", "i - anticipate", "i - see"
SC_51  - VC_95  e.g.  "chancellor - caution", "arafat - explain", "chernomyrdin - see"
SC_158 - VC_46  e.g.  "bolger - invite", "chairman - elect", "lott - accuse"
SC_53  - VC_22  e.g.  "that - choose", "structure - remain", "measure - like"
SC_184 - VC_95  e.g.  "bank - cite", "company - peg", "right - note"
SC_158 - VC_95  e.g.  "secretary - note", "official - detect", "official - speculate"
SC_53  - VC_45  e.g.  "deal - cost", "measure - protect", "extension - cover"
SC_184 - VC_64  e.g.  "syndicate - buy", "retailer - lease", "producer - buy"
SC_139 - VC_22  e.g.  "while - give", "point - give", "show - change"
SC_184 - VC_16  e.g.  "group - pay", "supplier - reduce", "syndicate - raise"
SC_184 - VC_11  e.g.  "unit - withhold", "utility - win", "it - refuse"
SC_127 - VC_79  e.g.  "them - miss", "bosnia - play", "management - regret"
SC_184 - VC_76  e.g.  "bank - suspend", "it - stop", "exchange - hold"
SC_56  - VC_74  e.g.  "share - tend", "stocks - settle", "future - mix"
SC_184 - VC_21  e.g.  "subsidiary - register", "unit - detail", "provider - post"
SC_127 - VC_95  e.g.  "bloc - say", "moscow - see", "they - trace"
SC_93  - VC_22  e.g.  "shanghai - place", "belgium - provide", "lebanon - put"
SC_13  - VC_25  e.g.  "reuter - verify",, "reuters - verify"
SC_51  - VC_15  e.g.  "clinton - applaud", "clarke - criticise", "minister - criticize"
SC_184 - VC_45  e.g.  "group - prevent", "carmakers - aim", "group - follow"
SC_51  - VC_79  e.g.  "clarke - know", "fernandez - want", "clinton - view"
SC_12  - VC_21  e.g.  "evidence - show", "study - report", "estimate - record"
SC_51  - VC_83  e.g.  "dole - liken", "coach - concede", "leader - rule"
SC_184 - VC_71  e.g.  "consortium - approve", "it - veto", "utility - present"
SC_51  - VC_11  e.g.  "mayor - win", "lebed - receive", "chancellor - win"
SC_70  - VC_53  e.g.  "rate - steady", "yen - grow", "naphtha - trade"
SC_127 - VC_76  e.g.  "state - participate", "both - hold", "team - postpone"
SC_127 - VC_11  e.g.  "majority - award", "star - win", "israel - accept"
SC_48  - VC_22  e.g.  "depreciation - give", "change - lack", "combination - set"
SC_184 - VC_79  e.g.  "developer - get", "bank - try", "service - afford"
SC_127 - VC_71  e.g.  "country - shelve", "paris - introduce", "partner - introduce"
SC_127 - VC_15  e.g.  "europe - praise", "nation - favour", "britain - challenge"
SC_70  - VC_22  e.g.  "price - shift", "sterling - return", "mark - give"
SC_96  - VC_71  e.g.  "meeting - table", "house - propose", "meeting - prepare"
SC_48  - VC_45  e.g.  "expansion - limit", "merger - enable", "change - involve"
SC_162 - VC_22  e.g.  "chinese - give", "chinese - keep", "chinese - move"
SC_184 - VC_4   e.g.  "service - strike", "supplier - conclude", "industry - structure"
SC_56  - VC_53  e.g.  "benchmark - stand", "escudo - fall", "index - firm"
SC_184 - VC_70  e.g.  "industry - mull", "bank - decide", "utility - monitor"
SC_41  - VC_53  e.g.  "volume - steady", "bankruptcy - total", "consumption - stand"
SC_51  - VC_76  e.g.  "samper - hold", "castro - delay", "lawyer - organise"
SC_14  - VC_22  e.g.  "increase - base", "surge - make", "pace - change"
```

7.2 Verb-Object Patterns

```
VC_46  - OC_75  e.g.  "brief - correspondent", "ask - television", "telephone - newspaper"
VC_22  - OC_131 e.g.  "work - fact", "put - account", "base - these"
VC_79  - OC_90  e.g.  "want - anything", "respect - what", "forget - something"
VC_22  - OC_184 e.g.  "establish - feeling", "need - connection", "remain - choice"
VC_22  - OC_189 e.g.  "exchange - policy", "provide - part", "exchange - vow"
VC_74  - OC_27  e.g.  "end - flat", "settle - cbot", "open - shade"
VC_95  - OC_131 e.g.  "forecast - those", "fear - many", "project - last"
VC_71  - OC_148 e.g.  "repeal - draft", "propose - text", "draft - mini-budget"
VC_79  - OC_131 e.g.  "mind - fact", "appreciate - fact", "miss - news"
VC_95  - OC_92  e.g.  "foresee - weakness", "forecast - continuation", "say - rise"
VC_4   - OC_47  e.g.  "strike - deal", "negotiate - lease", "sign - declaration"
VC_22  - OC_2   e.g.  "put - tally", "make - dive", "serve - place"
VC_22  - OC_170 e.g.  "use - flexibility", "offer - clue", "take - leverage"
VC_95  - OC_145 e.g.  "say - we", "explain - that", "add - syndicate"
VC_25  - OC_45  e.g.  "invent - story", "check - story", "invent - story"
VC_21  - OC_168 e.g.  "record - third-quarter", "record - earnings", "post - net"
VC_22  - OC_89  e.g.  "leave - us", "offer - doctor", "keep - speculator"
VC_22  - OC_121 e.g.  "switch - portfolio", "choose - portfolio", "apply - property"
VC_16  - OC_138 e.g.  "increase - discount", "cut - premium", "inflate - premium"
VC_95  - OC_103 e.g.  "expect - commodity", "see - security", "say - forex"
VC_22  - OC_98  e.g.  "have - confidence", "move - faith", "like - strength"
VC_76  - OC_150 e.g.  "halt - investigation", "initiate - negotiation", "kick - summit"
VC_95  - OC_22  e.g.  "trace - market", "say - firm", "attribute - trading"
VC_22  - OC_125 e.g.  "use - reference", "give - debut", "make - reference"
VC_22  - OC_22  e.g.  "keep - sell", "take - top", "become - banking"
VC_22  - OC_110 e.g.  "leave - better", "provide - hint", "remove - variation"
VC_22  - OC_81  e.g.  "place - government", "change - coalition", "offer - labour"
VC_46  - OC_89  e.g.  "inform - parent", "tell - parent", "telephone - him"
VC_95  - OC_90  e.g.  "cite - something", "say - brasil", "attribute - whatever"
VC_22  - OC_43  e.g.  "serve - initial", "take - job", "put - cash"
VC_34  - OC_64  e.g.  "except - percentage", "prod - data", "except - percentage"
VC_22  - OC_168 e.g.  "remain - gross", "make - loss", "give - result"
VC_16  - OC_155 e.g.  "trim - throughput", "double - consumption", "shave - likelihood"
VC_57  - OC_193 e.g.  "near - level", "widen - intraday", "approach - height"
VC_22  - OC_69  e.g.  "replace - selection", "have - judgement", "make - ruling"
VC_22  - OC_90  e.g.  "take - our", "serve - so", "change - what"
VC_21  - OC_92  e.g.  "report - easing", "reveal - growth", "show - contraction"
VC_28  - OC_26  e.g.  "reject - charge", "brush - accusation", "counter - claim"
VC_95  - OC_168 e.g.  "fear - cpi", "fear - surplus", "estimate - eps"
VC_22  - OC_155 e.g.  "change - deficit", "return - demand", "change - demand"
VC_22  - OC_148 e.g.  "send - proposal", "offer - bill", "provide - framework"
VC_85  - OC_197 e.g.  "suffer - complication", "tackle - competition", "survive - tough"
VC_22  - OC_104 e.g.  "need - backlog", "give - turnover", "lose - capacity"
VC_53  - OC_28  e.g.  "firm - percent", "climb - kobo", "jump - month-on-month"
VC_22  - OC_138 e.g.  "have - premium", "set - payout", "maintain - taxation"
VC_11  - OC_158 e.g.  "seek - authorisation", "win - order", "refuse - mandate"
VC_22  - OC_74  e.g.  "like - time", "lack - life", "need - delivery"
VC_22  - OC_197 e.g.  "remain - danger", "become - nightmare", "like - challenge"
VC_22  - OC_188 e.g.  "replace - operation", "use - sale", "send - investment"
VC_22  - OC_132 e.g.  "become - hedge", "serve - platform", "offer - entertainment"
```

7.3 Subject-Object Patterns

```
SC_185 - OC_131 e.g.  "nobody - ones", "bowler - most", "you - real"
SC_51  - OC_75  e.g.  "samper - reuters", "commander - television", "chief - radio"
SC_158 - OC_75  e.g.  "downer - abc", "peters - newspaper", "spokesman - audience"
SC_184 - OC_131 e.g.  "giant - his", "company - set", "bank - all"
SC_127 - OC_131 e.g.  "country - range", "community - some", "church - more"
SC_185 - OC_90  e.g.  "i - nothing", "i - something", "any - anything"
SC_184 - OC_168 e.g.  "group - monthly", "group - earnings", "gm - net"
SC_51  - OC_131 e.g.  "he - reading", "executive - business", "coach - most"
SC_184 - OC_121 e.g.  "city - holding", "subsidiary - fund", "business - power"
SC_13  - OC_45  e.g.  "reuters - item", "reuter - story", "reuter - story"
SC_184 - OC_188 e.g.  "branch - traffic", "monopoly - advertising", "unit - production"
SC_56  - OC_27  e.g.  "index - dax", "share - colombian", "eurodollar - lower"
SC_185 - OC_89  e.g.  "any - them", "everyone - them", "few - them"
SC_184 - OC_138 e.g.  "provider - fee", "industry - price", "telstra - dividend"
SC_184 - OC_47  e.g.  "firm - protocol", "company - memorandum", "province - agreement"
SC_51  - OC_165 e.g.  "mandela - representative", "governor - chancellor", "clinton - mayor"
SC_127 - OC_148 e.g.  "organisation - concept", "france - design", "italy - plan"
SC_127 - OC_90  e.g.  "employer - little", "staff - little", "they - so"
SC_51  - OC_81  e.g.  "kohl - country", "lawyer - side", "blair - institution"
SC_139 - OC_131 e.g.  "number - news", "one - his", "trade - due"
SC_184 - OC_148 e.g.  "province - regulation", "unit - version", "subsidiary - budget"
SC_51  - OC_150 e.g.  "chirac - mission", "governor - tour", "lawyer - talk"
SC_184 - OC_104 e.g.  "subsidiary - backlog", "company - income", "industry - income"
SC_127 - OC_189 e.g.  "will - policy", "they - initiative", "france - measure"
SC_184 - OC_92  e.g.  "business - pickup", "company - slowdown", "industry - slowdown"
SC_184 - OC_6   e.g.  "provider - location", "bank - location", "utility - site"
SC_184 - OC_155 e.g.  "business - spending", "gm - output", "bank - availability"
SC_51  - OC_90  e.g.  "kabila - what", "chirac - everything", "executive - little"
SC_184 - OC_22  e.g.  "producer - trading", "retailer - bank", "developer - market"
SC_184 - OC_189 e.g.  "bank - hold", "division - action", "enterprise - policy"
SC_51  - OC_148 e.g.  "lukashenko - rule", "premier - programme", "berisha - rule"
SC_53  - OC_131 e.g.  "system - business", "exercise - first", "step - those"
SC_96  - OC_148 e.g.  "committee - guideline", "lawmaker - amendment", "assembly - legislation"
SC_184 - OC_90  e.g.  "group - so", "boeing - what", "boeing - enough"
SC_184 - OC_96  e.g.  "unit - placement", "group - reorganization", "company - overhaul"
SC_184 - OC_132 e.g.  "unit - system", "unit - variety", "company - service"
SC_184 - OC_56  e.g.  "group - finding", "city - presentation", "company - invitation"
SC_51  - OC_89  e.g.  "she - voter", "netanyahu - him", "moi - them"
SC_185 - OC_92  e.g.  "you - decline", "others - improvement", "everyone - volatility"
SC_184 - OC_43  e.g.  "producer - billion", "it - initial", "it - amount"
SC_56  - OC_193 e.g.  "lev - low", "dax - peak", "yuan - level"
SC_70  - OC_27  e.g.  "gilt - lower", "differential - steady", "sterling - session"
SC_127 - OC_47  e.g.  "republic - multi", "britain - treaty", "member - decree"
SC_127 - OC_89  e.g.  "community - you", "farmer - us", "bloc - them"
SC_184 - OC_87  e.g.  "bank - share", "it - share", "it - yuan"
SC_184 - OC_81  e.g.  "bt - team", "bank - minority", "firm - department"
SC_185 - OC_189 e.g.  "many - aim", "any - action", "everyone - part"
SC_51  - OC_189 e.g.  "fernandez - policy", "general - aim", "albright - line"
SC_184 - OC_89  e.g.  "developer - them", "exchange - shareholder", "trust - her"
SC_185 - OC_145 e.g.  "we - economics", "we - they", "people - he"
```

7.4 Word-Word Patterns

7.4.1 Language Phrase Patterns

```
bear (VC_32)       - gift (OC_154)
smoke (VC_9)       - pack (OC_15)
wear (VC_7)        - wire (OC_194)
sound (VC_39)      - horn (OC_65)
breathe (VC_9)     - sigh (OC_129)
perform (VC_94)    - abortion (OC_123)
radio (VC_19)      - back (OC_25)
heap (VC_100)      - scorn (OC_129)
amass (VC_30)      - fortune (OC_124)
cast (VC_3)        - eye (OC_73)
```

7.4.2 World Patterns

```
texas (SC_180)     - execute (VC_93)
tyson (SC_23)      - bite (VC_19)
fighter (SC_167)   - scramble (VC_89)
repair (VC_72)     - damage (OC_153)
surgery (SC_142)   - knee (OC_149)
carcass (SC_132)   - disease (OC_84)
hearing (SC_2)     - witness (OC_57)
hole (SC_60)       - plug (OC_180)
jail (SC_145)      - suspect (OC_109)
publisher (SC_98)  - book (OC_113)
territory (SC_105) - attack (OC_115)
dividend (SC_129)  - stockholder (OC_112)
doctor (SC_182)    - pinch (OC_44)
god (SC_60)        - america (OC_86)
```

7.4.3 Patterns Resulting from Parser Misclassification

```
soy (SC_180)       - crush (VC_10)
snap (VC_37)       - shot (OC_67)
mortgage (VC_87)   - guide (OC_37)
sport (VC_13)      - broadcasting (OC_46)
retire (VC_72)     - batter (OC_76)
colony (SC_105)    - former (OC_38)
england (SC_76)    - former (OC_38)
```

7.4.4 Patterns Specific to the Corpus

```
000s (SC_146)      - except (VC_34)
e-mail (VC_34)     - derivativesdesk (OC_129)
```

8 Appendix C - Inter-Clause Patterns

<u>Note</u>: for convenience, we use **'-(*t*)-'** to represents the fact that the second part of the pattern followed the first part within the next *t* clauses.

8.1 Patterns within Three Clauses (*t* = 3)

Due to lack of space, we present here the ten most significant patterns for each possible combination of cluster categories. We omit the cluster repeat patterns when listing the verb-verb, subject-subject, and object-object patterns. Almost all of the repeat patterns are the most significant instances of their pattern template, and were discussed briefly in the results section. Patterns stemming from the Reuters docking manifests are marked in **bold** font. Three examples are provided for each pattern.

SC_64	-(3)-	SC_79	e.g.	"0830 - ex", "0930 - ex", "0830 - ex"
SC_102	**-(3)-**	**SC_128**	**e.g.**	**"steer - kansas", "heifer - gluten", "steer - okla"**
SC_17	-(3)-	SC_56	e.g.	"broker - dax", "observer - peso", "economist - future"
SC_54	-(3)-	SC_167	e.g.	"gunman - troop", "gunmen - force", "eta - navy"
SC_14	-(3)-	SC_56	e.g.	"increase - lumber", "recovery - benchmark", "dip - rupee"
SC_139	-(3)-	SC_56	e.g.	"centre - real", "offer - yuan", "rest - index"
SC_69	**-(3)-**	**SC_157**	**e.g.**	**"loader - 1", "bulk - loading", "sulphur - container"**
SC_17	-(3)-	SC_70	e.g.	"analyst - drachma", "insider - yen", "broker - dollar"
SC_46	-(3)-	SC_167	e.g.	"raid - police", "raid - rebel", "shooting - guard"
SC_167	-(3)-	SC_54	e.g.	"fighter - commando", "army - militiamen", "soldier - gunmen"

SC_157	**-(3)-**	**VC_68**	**e.g.**	**"loading - empty", "1 - bulk", "cement - coal"**
SC_69	**-(3)-**	**VC_68**	**e.g.**	**"tonne - load", "tonne - empty", "loader - wire"**
SC_167	-(3)-	VC_40	e.g.	"troop - shoot", "army - arrest", "navy - wound"
SC_56	-(3)-	VC_74	e.g.	"future - fix", "stocks - finish", "mos - open"
SC_99	-(3)-	VC_36	e.g.	"blast - sweep", "earthquake - rattle", "landslide - ravage"
SC_167	-(3)-	VC_5	e.g.	"soldier - bombard", "tank - control", "military - attack"
SC_54	-(3)-	VC_40	e.g.	"youth - kill", "gunmen - kill", "eta - shoot"
SC_185	-(3)-	VC_79	e.g.	"someone - get", "someone - accomplish", "anyone - appreciate"
SC_193	-(3)-	VC_74	e.g.	"crude - settle", "palladium - mix", "back - settle"
SC_56	-(3)-	VC_53	e.g.	"peso - stand", "lumber - total", "jgbs - trade"

SC_64	-(3)-	OC_77	e.g.	"0930 - u.k.", "0830 - u.k.", "0645 - u.k."
SC_157	**-(3)-**	**OC_156**	**e.g.**	**"2 - container", "fertiliser - vessel", "cement - 4"**
SC_157	**-(3)-**	**OC_130**	**e.g.**	**"container - flour", "loading - berth", "4 - cement"**
SC_69	**-(3)-**	**OC_130**	**e.g.**	**"loader - cement", "loader - berth", "sulphur - discharge"**
SC_56	-(3)-	OC_27	e.g.	"share - lme", "dlr - lower", "stocks - bourse"
SC_167	-(3)-	OC_152	e.g.	"troop - kabul", "military - stronghold", "rebel - grozny"
SC_138	-(3)-	OC_148	e.g.	"legislation - package", "rule - version", "treaty - package"
SC_76	-(3)-	OC_166	e.g.	"chelsea - 4-0", "bolivia - 2-1", "newcastle - 5-1"
SC_195	-(3)-	OC_85	e.g.	"islamabad - pakistan", "iraq - burma", "baghdad - washington"
SC_167	-(3)-	OC_120	e.g.	"soldier - rwandan", "police - taleban", "police - zairean"

VC_68	**-(3)-**	**SC_157**	**e.g.**	**"steel - coal", "coal - cement", "iron - 2"**
VC_68	**-(3)-**	**SC_69**	**e.g.**	**"steel - liquid", "bag - bulk", "berth - sulphur"**
VC_74	-(3)-	SC_56	e.g.	"finish - benchmark", "skid - future", "nudge - stock"
VC_5	-(3)-	SC_167	e.g.	"retake - troop", "bomb - rebel", "seize - rebel"
VC_40	-(3)-	SC_167	e.g.	"kill - militia", "injure - troop", "disperse - troop"
VC_36	-(3)-	SC_99	e.g.	"shake - blast", "sweep - flood", "grip - typhoon"
VC_40	-(3)-	SC_54	e.g.	"kidnap - bomber", "assault - gunmen", "injure - youth"
VC_79	-(3)-	SC_185	e.g.	"feel - me", "appreciate - me", "get - we"

VC_53	-(3)-	SC_56	e.g.	"jump - rupee", "ease - jgbs", "sag - future"
VC_74	-(3)-	SC_193	e.g.	"end - cattle", "skid - soybean", "finish - cocoa"

VC_68	**-(3)-**	**VC_56**	**e.g.**	**"bag - bunker", "discharge - copper", "bulk - copper"**
VC_56	**-(3)-**	**VC_68**	**e.g.**	**"sugar - bag", "oil - bag", "sugar - wire"**
VC_6	-(3)-	VC_40	e.g.	"shatter - arrest", "smash - detain", "wreck - injure"
VC_5	-(3)-	VC_40	e.g.	"invade - shoot", "overrun - slaughter", "occupy - displace"
VC_53	-(3)-	VC_74	e.g.	"fall - settle", "edge - expire", "climb - settle"
VC_52	-(3)-	VC_40	e.g.	"patrol - arrest", "guard - shoot", "surround - kidnap"
VC_39	-(3)-	VC_95	e.g.	"downplay - fear", "upgrade - project", "downplay - say"
VC_1	-(3)-	VC_40	e.g.	"deploy - wound", "ally - kill", "fire - injure"
VC_68	-(3)-	VC_67	e.g.	"bulk - ship", "load - supply", "truck - process"
VC_74	-(3)-	VC_53	e.g.	"finish - steady", "close - spike", "advance - plunge"

VC_68	**-(3)-**	**OC_130**	**e.g.**	**"steel - discharger", "bulk - discharge", "discharge - loader"**
VC_68	**-(3)-**	**OC_156**	**e.g.**	**"discharge - 2", "discharge - loading", "steel - wharf"**
VC_74	-(3)-	OC_27	e.g.	"mix - toronto", "expire - spot", "settle - dax"
VC_97	**-(3)-**	**OC_196**	**e.g.**	**"cash - soybean", "spring - rice", "mill - rice"**
VC_40	-(3)-	OC_146	e.g.	"arrest - student", "shoot - wife", "kill - boy"
VC_68	**-(3)-**	**OC_194**	**e.g.**	**"load - scrap", "wire - timber", "steel - timber"**
VC_71	-(3)-	OC_148	e.g.	"vote - plan", "violate - budget", "submit - framework"
VC_5	-(3)-	OC_152	e.g.	"capture - kabul", "attack - kinshasa", "attack - kinshasa"
VC_4	-(3)-	OC_47	e.g.	"arrange - pact", "strike - pact", "structure - deal"
VC_68	**-(3)-**	**OC_196**	**e.g.**	**"load - wheat", "cement - wheat", "load - crude"**

OC_77	-(3)-	SC_64	e.g.	"manuf - 0645", "u.k. - 0800", "ita - 0800"
OC_156	**-(3)-**	**SC_157**	**e.g.**	**"vessel - fertiliser", "wharf - fertiliser", "vessel - coal"**
OC_130	**-(3)-**	**SC_157**	**e.g.**	**"cement - 2", "flour - loading", "fertiliser - loading"**
OC_126	**-(3)-**	**SC_128**	**e.g.**	**"cattle - plain", "cattle - okla", "bulk - gluten"**
OC_130	**-(3)-**	**SC_69**	**e.g.**	**"fio - sulphur", "cement - loader", "cement - loader"**
OC_27	-(3)-	SC_56	e.g.	"flat - indices", "kerb - index", "nasdaq - index"
OC_77	-(3)-	SC_79	e.g.	"manuf - ex", "m4 - indust", "u.k. - non-farm"
OC_152	-(3)-	SC_167	e.g.	"kabul - rebel", "camp - soldier", "camp - rebel"
OC_166	-(3)-	SC_76	e.g.	"3-2 - juventus", "2-0 - england", "3-0 - liverpool"
OC_117	-(3)-	SC_19	e.g.	"landslide - foe", "election - liberal", "seat - ally"

OC_156	**-(3)-**	**VC_68**	**e.g.**	**"container - iron", "1 - bulk", "vessel - truck"**
OC_130	**-(3)-**	**VC_68**	**e.g.**	**"terminal - discharge", "flour - bag", "loader - berth"**
OC_27	-(3)-	VC_74	e.g.	"gmt - expire", "cme - finish", "touch - mix"
OC_146	-(3)-	VC_40	e.g.	"palestinian - slay", "settler - kill", "palestinian - assault"
OC_196	**-(3)-**	**VC_97**	**e.g.**	**"rice - cash", "crude - condition", "maize - yellow"**
OC_196	**-(3)-**	**VC_68**	**e.g.**	**"aluminium - load", "corn - coal", "copper - empty"**
OC_152	-(3)-	VC_5	e.g.	"town - overrun", "arbil - attack", "village - capture"
OC_59	**-(3)-**	**VC_68**	**e.g.**	**"chicken - steel", "steel - coal", "steel - bag"**
OC_194	**-(3)-**	**VC_68**	**e.g.**	**"pipe - steel", "scrap - load", "scrap - cement"**
OC_148	-(3)-	VC_71	e.g.	"plan - pass", "regulation - violate", "law - debate"

OC_130	-(3)-	OC_156	e.g.	"discharger - wharf", "terminal - wharf", "fertiliser - 1"
OC_194	**-(3)-**	**OC_156**	**e.g.**	**"timber - container", "timber - loading", "pipe - container"**
OC_166	-(3)-	OC_34	e.g.	"3-1 - final", "3-0 - final", "3-0 - premier"
OC_142	-(3)-	OC_166	e.g.	"game - 5-0", "soccer - 4-1", "football - 1-0"
OC_27	-(3)-	OC_54	e.g.	"dealings - rose", "lower - finish", "gmt - ease"
OC_34	-(3)-	OC_166	e.g.	"semifinal - 2-1", "final - 3-0", "champion - 2-0"
OC_166	-(3)-	OC_142	e.g.	"1-0 - golf", "2-0 - role", "2-1 - tennis"
OC_152	-(3)-	OC_120	e.g.	"town - tutsis", "village - islamist", "kisangani - zairean"
OC_175	-(3)-	OC_196	e.g.	"carryover - soybean", "harvest - beef", "crop - gasoline"
OC_144	-(3)-	OC_34	e.g.	"slalom - results", "championship - final", "cup - champion"

9 Appendix D – Complex Patterns

The results presented here are the top scoring patterns for each type of anchoring system. More generalized patterns are ranked higher than more specific ones, regardless of their significance score. Within each level of generalization, ranking is according to the statistical significance. Only patterns with more than three occurrences are presented. Actual examples from the data follow each pattern. Where possible (i.e. there were at least three *different* instances in the data), three examples are given. In some cases, more examples were shown, to present the diversity of the words within the clusters. In the examples, each word in the pattern is followed by its cluster, in parentheses. Clusters are marked as 'C_*n*' or 'C*n*' where *n* is the cluster index. The category of the cluster should be inferred from its position in the clause. Patterns resulting from Reuters' docking manifests are marked in **bold** font (see final paragraph in section 4).

9.1 Patterns with Subject - Subject Anchor

```
x, C_68, C_196 -(10)- x, C_71, C_156 Encountered 6 instances:
steel     (C157), discharge (C068), rice      (C196) -- steel     (C157), scrap     (C070), vessel    (C156)
vessel    (C026), berth     (C068), maize     (C196) -- vessel    (C026), draft     (C070), 2         (C156)
vessel    (C026), berth     (C068), crude     (C196) -- vessel    (C026), draft     (C070), 2         (C156)

x, C_36, C_7 -(10)- x, C_57, C_85  Encountered 5 instances:
storm     (C099), lash      (C036), province  (C007) -- storm     (C099), cross     (C057), cuba      (C085)
quake     (C099), shake     (C036), city      (C007) -- quake     (C099), hit       (C057), iran      (C085)
earthquake(C099), jolt      (C036), city      (C007) -- earthquake(C099), hit       (C057), iran      (C085)

x, C_56, C_155 -(10)- x, C_67, C_141  Encountered 17 instances:
malaysia  (C093), rubber (C056), output      (C155) -- malaysia  (C093), import    (C067), tonne     (C140)
mexico    (C118), sugar  (C056), output      (C155) -- mexico    (C118), export    (C067), tonne     (C140)
syria     (C195), gas    (C056), output      (C155) -- syria     (C195), produce   (C067), bpd       (C140)

x, C_40, C_165 -(10)- x, C_52, C_152  Encountered 4 instances:
police    (C167), arrest    (C040), leader    (C165) -- police    (C167), search    (C052), mosque    (C152)
police    (C167), detain    (C040), leader    (C165) -- police    (C167), search    (C052), mosque    (C152)
police    (C167), arrest    (C040), member    (C165) -- police    (C167), raid      (C052), enclave   (C152)

x, C_11, C_196 -(10)- x, C_65, C_83   Encountered 4 instances:
pakistan  (C093), seek      (C011), wheat     (C196) -- pakistan  (C093), tender    (C065), grade     (C083)
india     (C093), receive   (C011), crude     (C196) -- india     (C093), contract  (C065), cargo     (C083)

x, C_68, C_156 -(10)- x, freeze, C_59 Encountered 12 instances:
2         (C157), bag       (C068), loading   (C156) -- 2         (C157), freeze    (C011), meat      (C059)
1         (C157), discharge (C068), container (C156) -- 1         (C157), freeze    (C011), chicken   (C059)
2         (C157), bag       (C068), loading   (C156) -- 2         (C157), freeze    (C011), meat      (C059)

x, C_26, pct -(10)- x, C_11, C_138  Encountered 68 instances:
avg       (C062), price     (C026), pct       (C028) -- avg       (C062), accept    (C011), price     (C138)
avg       (C062), yield     (C026), pct       (C028) -- avg       (C062), accept    (C011), price     (C138)
t-bill    (C077), rate      (C026), pct       (C028) -- t-bill    (C077), issue     (C011), discount  (C138)

x, feed, C_130 -(10)- x, C_68, C_196  Encountered 4 instances:
3         (C157), feed      (C054), meal      (C130) -- 3         (C157), bag       (C068), sugar     (C196)
2         (C157), feed      (C054), meal      (C130) -- 2         (C157), bag       (C068), sugar     (C196)
vessel    (C026), feed      (C054), loader    (C130) -- vessel    (C026), bulk      (C068), sulphur   (C196)
```

```
x, C_36, C_7 -(10)- x, cause, C_153  Encountered 13 instances:
earthquake(C099), shake     (C036), area      (C007) -- earthquake(C099), cause     (C023), casualty  (C153)
earthquake(C099), jolt      (C036), region    (C007) -- earthquake(C099), cause     (C023), damage    (C153)
earthquake(C099), shake     (C036), island    (C007) -- earthquake(C099), cause     (C023), injury    (C153)
quake     (C099), jolt      (C036), province  (C007) -- quake     (C099), cause     (C023), damage    (C153)
earthquake(C099), jolt      (C036), city      (C007) -- earthquake(C099), cause     (C023), damage    (C153)
tremor    (C099), shake     (C036), coast     (C007) -- tremor    (C099), cause     (C023), damage    (C153)

x, block, C_192 -(10)- x, C_12, C_182  Encountered 4 instances:
police    (C167), block     (C076), road      (C192) -- police    (C167), man       (C012), checkpoin (C182)
police    (C167), block     (C076), street    (C192) -- police    (C167), dismantle (C012), cordon    (C182)
worker    (C116), block     (C076), road      (C192) -- worker    (C116), erect     (C012), barricade (C182)
union     (C127), block     (C076), road      (C192) -- union     (C127), man       (C012), roadblock (C182)

x, freeze, C_59 -(10)- x, C_68, C_156  Encountered 4 instances:
1         (C157), freeze    (C011), fish      (C059) -- 1         (C157), anchor    (C068), waiting   (C156)
vessel    (C026), freeze    (C011), meat      (C059) -- vessel    (C026), berth     (C068), wharf     (C156)
2         (C157), freeze    (C011), chicken   (C059) -- 2         (C157), bag       (C068), vessel    (C156)

x, C_36, C_7 -(10)- x, damage, C_39  Encountered 4 instances:
quake     (C099), jolt      (C036), province  (C007) -- quake     (C099), damage    (C063), building  (C039)
explosion (C099), rock      (C036), city      (C007) -- explosion (C099), damage    (C063), building  (C039)
quake     (C099), devastate (C036), area      (C007) -- quake     (C099), damage    (C063), home      (C039)

x, C_11, delay -(10)- x, C_29, C_94  Encountered 9 instances:
italy     (C127), seek      (C011), delay     (C032) -- italy     (C127), meet      (C029), criteria  (C094)
germany   (C127), seek      (C011), delay     (C032) -- germany   (C127), meet      (C029), criteria  (C094)
germany   (C127), accept    (C011), delay     (C032) -- germany   (C127), fulfil    (C029), criteria  (C094)
germany   (C127), accept    (C011), delay     (C032) -- germany   (C127), overshoot (C029), target    (C094)

x, break, C_10 -(10)- x, C_11, C_74  Encountered 4 instances:
israel    (C127), break     (C057), ground    (C010) -- israel    (C127), accept    (C011), pause     (C074)
she       (C051), break     (C057), serve     (C010) -- she       (C051), win       (C011), time      (C074)

x, issue, C_87 -(10)- x, C_76, C_91  Encountered 5 instances:
russia    (C093), issue     (C011), bond      (C087) -- russia    (C093), launch    (C076), eurobond  (C091)
russia    (C093), issue     (C011), bond      (C087) -- russia    (C093), launch    (C076), eurobond  (C091)
eib       (C118), issue     (C011), bond      (C087) -- eib       (C118), launch    (C076), eurobond  (C091)
russia    (C093), issue     (C011), bond      (C087) -- russia    (C093), launch    (C076), eurobond  (C091)
russia    (C093), issue     (C011), rouble    (C087) -- russia    (C093), plan      (C076), eurobonds (C091)

x, C_57, C_7 -(10)- x, cause, C_153  Encountered 7 instances:
storm     (C099), hit       (C057), coast     (C007) -- storm     (C099), cause     (C023), damage    (C153)
cyclone   (C099), near      (C057), coast     (C007) -- cyclone   (C099), cause     (C023), damage    (C153)
cyclone   (C099), near      (C057), coast     (C007) -- cyclone   (C099), cause     (C023), damage    (C153)
earthquake(C099), hit       (C057), northwest (C007) -- earthquake(C099), cause     (C023), damage    (C153)
quake     (C099), hit       (C057), northwest (C007) -- quake     (C099), cause     (C023), damage    (C153)
quake     (C099), hit       (C057), northwest (C007) -- quake     (C099), cause     (C023), casualty  (C153)
earthquake(C099), hit       (C057), city      (C007) -- earthquake(C099), cause     (C023), damage    (C153)

x, win, C_142 -(10)- x, C_10, C_89  Encountered 4 instances:
side      (C127), win       (C011), game      (C142) -- side      (C127), crush     (C010), them      (C089)
i         (C185), win       (C011), match     (C142) -- i         (C185), beat      (C010), her       (C089)
i         (C185), win       (C011), match     (C142) -- i         (C185), beat      (C010), him       (C089)

x, see, C_196 -(10)- x, C_57, C_141  Encountered 4 instances:
export    (C041), see       (C095), rice      (C196) -- export    (C041), hit       (C057), tonne     (C141)
export    (C041), see       (C095), beef      (C196) -- export    (C041), hit       (C057), tonne     (C141)
import    (C041), see       (C095), sugar     (C196) -- import    (C041), touch     (C057), tonne     (C141)
crop      (C086), see       (C095), cotton    (C196) -- crop      (C086), near      (C057), bale      (C141)

x, C_84, rose -(10)- x, C_21, C_4  Encountered 4 instances:
automaker (C039), dominate  (C084), rose   (C054) -- automaker (C039), post   (C021), gain      (C004)
automaker (C039), dominate  (C084), rose   (C054) -- automaker (C039), post   (C021), gain      (C004)
automaker (C039), dominate  (C084), rose   (C054) -- automaker (C039), post   (C021), gain      (C004)
sales     (C041), lead      (C084), rose   (C054) -- sales     (C041), show   (C021), recovery  (C004)

x, love, C_46 -(10)- x, C_79, C_90  Encountered 4 instances:
i         (C185), love    (C061), campaigning (C046) -- i         (C185), know      (C079), everythin (C090)
norman    (C151), love    (C061), fishing     (C046) -- norman    (C151), do        (C079), thing     (C090)
i         (C185), love    (C061), racing      (C046) -- i         (C185), do        (C079), what      (C090)
i         (C185), love    (C061), playing     (C046) -- i         (C185), play      (C079), well      (C090)
```

9.2 Patterns with Verb - Verb Anchor

```
C_157, x, C_196 -(10)- C_26, x, C_130  Encountered 13 instances:
1          (C157), bag       (C068), rice       (C196) -- vessel    (C026), bag   (C068), cement     (C130)
1          (C157), bag       (C068), sugar      (C196) -- cargo     (C026), bag   (C068), flour      (C130)
2          (C157), bag       (C068), maize      (C196) -- vessel    (C026), bag   (C068), fertiliser (C130)

C_26, x, C_130 -(10)- C_157, x, C_196  Encountered 17 instances:
vessel     (C026), bulk      (C068), fertiliser (C130) -- 1          (C157), bulk   (C068), corn       (C196)
vessel     (C026), bulk      (C068), fertiliser (C130) -- 2          (C157), bulk   (C068), wheat      (C196)
vessel     (C026), bag       (C068), fertiliser (C130) -- 2          (C157), bag    (C068), rice       (C196)

C_157, x, C_130 -(10)- C_26, x, C_196  Encountered 17 instances:
2          (C157), bag       (C068), cement     (C130) -- vessel    (C026), bag    (C068), sugar      (C196)
1          (C157), bag       (C068), cement     (C130) -- vessel    (C026), bag    (C068), rice       (C196)
1          (C157), bag       (C068), fertilise  (C130) -- cargo     (C026), bag    (C068), sugar      (C196)

C_26, x, C_196 -(10)- C_157, x, C_130  Encountered 7 instances:
vessel     (C026), bag       (C068), rice       (C196) -- 1         (C157), bag    (C068), fertiliser (C130)
cargo      (C026), bag       (C068), sugar      (C196) -- steel     (C157), bag    (C068), flour      (C130)
vessel     (C026), bag       (C068), rice       (C196) -- 2         (C157), bag    (C068), fertiliser (C130)

C_192, x, C_54 -(10)- C_8, x, C_27  Encountered 7 instances:
kl         (C192), share     (C054), end        (C054) -- malaysian (C008), share     (C054), lower     (C027)
slovenian  (C192), share     (C054), ease       (C054) -- singapore (C008), share     (C054), steady    (C027)
kl         (C192), share     (C054), end        (C054) -- malaysian (C008), share     (C054), lower     (C027)

C_56, x, C_42 -(10)- C_17, x, C_87  Encountered 9 instances:
share      (C056), see       (C095), slovenia   (C042) -- trader    (C017), see       (C095), share     (C087)
stock      (C056), see       (C095), mexico     (C042) -- trader    (C017), see       (C095), stock     (C087)
stocks     (C056), see       (C095), mexico     (C042) -- dealer    (C017), see       (C095), stock     (C087)

C_167, x, C_58 -(10)- C_54, x, C_146  Encountered 11 instances:
rebel      (C167), kill      (C040), two        (C058) -- guerrilla (C054), kill      (C040), people    (C146)
bodyguard  (C167), kill      (C040), six        (C058) -- guerrilla (C054), kill      (C040), soldier   (C146)
squad      (C167), kill      (C040), five       (C058) -- gunmen    (C054), kill      (C040), man       (C146)
rebel      (C167), kill      (C040), five       (C058) -- guerrilla (C054), kill      (C040), people    (C146)
army       (C167), kill      (C040), fighter    (C058) -- guerrilla (C054), kill      (C040), soldier   (C146)
policemen  (C167), kill      (C040), seven      (C058) -- attacker  (C054), kill      (C040), policemen (C146)

C_158, x, C_22 -(10)- C_17, x, C_43  Encountered 33 instances:
source     (C158), say       (C095), market     (C022) -- trader    (C017), say       (C095), money     (C043)
official   (C158), say       (C095), bank       (C022) -- dealer    (C017), say       (C095), amount    (C043)
official   (C158), say       (C095), company    (C022) -- economist (C017), say       (C095), benefit   (C043)

C_93, x, C_196 -(10)- C_127, x, C_141  Encountered 13 instances:
argentina (C093), sell     (C064), wheat     (C196) -- france     (C127), sell     (C064), tonne     (C141)
brazil    (C093), sell     (C064), coffee    (C196) -- government (C127), sell     (C064), bag       (C141)
india     (C093), produce  (C067), aluminium (C196) -- country    (C127), produce  (C067), tonne     (C141)
india     (C093), allocate (C008), wheat     (C196) -- government (C127), allocate (C008), tonne     (C141)
china     (C093), buy      (C064), rubber    (C196) -- government (C127), buy      (C064), tonne     (C141)
india     (C093), produce  (C067), zinc      (C196) -- country    (C127), produce  (C067), tonne     (C141)

C_93, x, C_196 -(10)- C_121, x, C_131  Encountered 6 instances:
india      (C093), buy       (C064), crude      (C196) -- ioc       (C121), buy       (C064), any       (C131)
china      (C093), buy       (C064), soyoil     (C196) -- adm       (C121), buy       (C064), each      (C131)
taiwan     (C093), buy       (C064), soybean    (C196) -- adm       (C121), buy       (C064), lot       (C131)

C_158, x, C_89 -(10)- C_30, x, C_75  Encountered 7 instances:
chairman   (C158), tell   (C046), shareholder (C089) -- officer   (C030), tell   (C046), reuters        (C075)
official   (C158), tell   (C046), analyst     (C089) -- daughter  (C030), tell   (C046), newspaper      (C075)
official   (C158), tell   (C046), him         (C089) -- man       (C030), tell   (C046), correspondent  (C075)

C_54, x, C_146 -(10)- C_99, x, C_58  Encountered 5 instances:
bomber     (C054), kill    (C040), people      (C146) -- bomb      (C099), kill      (C040), american  (C058)
guerrilla  (C054), kill    (C040), civilian    (C146) -- explosion (C099), kill      (C040), six       (C058)
terrorist  (C054), kill    (C040), people      (C146) -- bomb      (C099), kill      (C040), score     (C058)
settler    (C054), kill    (C040), palestinian (C146) -- blast     (C099), kill      (C040), israeli   (C058)
```

9.3 Patterns with Object – Object Anchor

```
C_104, C_22, x -(10)- C_188, C_53, x  Encountered 5 instances:
unibanka  (C104), make      (C022), lat       (C028) -- factory   (C188), trade     (C053), lat       (C028)
unibanka  (C104), turn      (C022), lat       (C028) -- plant     (C188), trade     (C053), lat       (C028)
unibanka  (C104), turn      (C022), lat       (C028) -- factory   (C188), climb     (C053), lat       (C028)
hansapank (C104), turn      (C022), kroons    (C028) -- plant     (C188), slip      (C053), kroons    (C028)
hoiupank  (C104), lose      (C022), kroons    (C028) -- factory   (C188), trade     (C053), kroons    (C028)

C_26, C_68, x -(10)- C_157, C_71, x  Encountered 4 instances:
cargo     (C026), empty     (C068), vessel    (C156) -- 4         (C157), map       (C071), vessel    (C156)
tanker    (C026), coal      (C068), due       (C131) -- container (C157), scrap     (C071), due       (C131)
tanker    (C026), coal      (C068), due       (C131) -- container (C157), scrap     (C071), due       (C131)
vessel    (C026), berth     (C068), general   (C165) -- loading   (C157), scrap     (C071), general   (C165)

C_193, C_83, x -(10)- C_139, C_53, x  Encountered 10 instances:
copper    (C193), hop      (C083), resistance (C197) -- level   (C139), firm      (C053), resistance (C197)
natgas    (C193), call     (C083), lower      (C027) -- last    (C139), trade     (C053), lower     (C027)

ship, C_59, x -(10)- C_135, C_68, x  Encountered 114 instances:
ship      (C026), sail      (C059), five      (C058) -- four      (C135), berth     (C068), five      (C058)
ship      (C026), sail      (C059), five      (C058) -- six       (C135), berth     (C068), five      (C058)
ship      (C026), dock      (C059), sugar     (C196) -- 15        (C135), bag       (C068), sugar     (C196)

C_18, post, x -(10)- C_14, C_45, x  Encountered 70 instances:
u.s.      (C018), post      (C021), price     (C138) -- increase  (C014), bring     (C045), price     (C138)
u.s.      (C018), post      (C021), price     (C138) -- news      (C014), further   (C045), price     (C138)
u.s.      (C018), post      (C021), price     (C138) -- rise      (C014), bring     (C045), price     (C138)
us        (C018), post      (C021), price     (C138) -- increase  (C014), bring     (C045), price     (C138)

bank, C_100, x -(10)- C_139, C_65, x  Encountered 17 instances:
bank      (C184), prime     (C100), rate      (C138) -- term      (C139), deposit   (C065), rate      (C138)
bank      (C184), steer     (C100), rate      (C138) -- bid       (C139), auction   (C065), rate      (C138)

C_70, C_74, x -(10)- interbank, C_83, x  Encountered 24 instances:
rate      (C070), end       (C074), money     (C043) -- interbank (C175), call      (C083), money     (C043)
rate      (C070), finish    (C074), money     (C043) -- interbank (C175), call      (C083), money     (C043)
rate      (C070), close     (C074), money     (C043) -- interbank (C175), call      (C083), money     (C043)
currency  (C070), mix       (C074), money     (C043) -- interbank (C175), term      (C083), money     (C043)
rate      (C070), close     (C074), money     (C043) -- interbank (C175), call      (C083), money     (C043)

C_128, feed, x -(10)- C_102, C_57, x  Encountered 9 instances:
nebraska  (C128), feed      (C054), cattle    (C126) -- well      (C102), test      (C057), cattle    (C126)
nebraska  (C128), feed      (C054), steady    (C027) -- heifer    (C102), near      (C057), steady    (C027)
okla      (C128), feed      (C054), cattle    (C126) -- well      (C102), test      (C057), cattle    (C126)

C_93, beat, x -(10)- C_16, C_77, x  Encountered 4 instances:
canada    (C093), beat      (C010), 3-1       (C166) -- 2-0       (C016), match     (C077), 3-1       (C166)
colombia  (C093), beat      (C010), 2-0       (C166) -- 1-0       (C016), match     (C077), 2-0       (C166)
colombia  (C093), beat      (C010), 4-1       (C166) -- 2-0       (C016), round     (C077), 4-1       (C166)
australia (C093), beat      (C010), 2-0       (C166) -- 1-0       (C016), match     (C077), 2-0       (C166)

C_26, C_53, x -(10)- C_18, handle, x  Encountered 9 instances:
cargo     (C026), trade     (C053), airport   (C039) -- traffic   (C018), handle    (C067), airport   (C039)
freight   (C026), total     (C053), tonne     (C141) -- airport   (C018), handle    (C067), tonne     (C141)
cargo     (C026), total     (C053), kg        (C141) -- airport   (C018), handle    (C067), kg        (C141)
cargo     (C026), total     (C053), tonne     (C141) -- airport   (C018), handle    (C067), tonne     (C141)
cargo     (C026), total     (C053), tonne     (C141) -- airport   (C018), handle    (C067), tonne     (C141)
```

9.4 Patterns with Subject - Object Anchor

```
x, C_68, C_156 -(10)- C_157, C_56, x  Encountered 6 instances:
vessel    (C026), berth     (C068), wharf    (C156) -- loading    (C157), metal     (C056), vessel    (C156)
vessel    (C026), cement    (C068), loading  (C156) -- fertiliser (C157), metal     (C056), vessel    (C156)
vessel    (C026), berth     (C068), 1        (C156) -- 3          (C157), copper    (C056), vessel    (C156)

x, C_15, C_152 -(10)- C_167, C_40, x  Encountered 6 instances:
militant  (C054), oppose    (C015), mosque    (C152) -- police     (C167), detain    (C040), militant  (C109)
soldier   (C167), abandon   (C015), town      (C152) -- rebel      (C167), kill      (C040), soldier   (C146)
refugee   (C021), abandon   (C015), camp      (C152) -- troop      (C167), shoot     (C040), refugee   (C164)

x, berth, C_58 -(10)- C_135, C_56, x  Encountered 6 instances:
vessel    (C026), berth     (C068), three     (C058) -- two        (C135), oil       (C056), vessel    (C156)
vessel    (C026), berth     (C068), three     (C058) -- four       (C135), oil       (C056), vessel    (C156)
vessel    (C026), berth     (C068), three     (C058) -- three      (C135), oil       (C056), vessel    (C156)

x, berth, C_156 -(10)- C_135, C_56, x  Encountered 4 instances:
vessel    (C026), berth     (C068), 2         (C156) -- four       (C135), oil       (C056), vessel    (C156)
vessel    (C026), berth     (C068), 1         (C156) -- three      (C135), oil       (C056), vessel    (C156)
vessel    (C026), berth     (C068), waiting   (C156) -- two        (C135), copper    (C056), vessel    (C156)

x, C_68, C_156 -(10)- one, C_56, x  Encountered 6 instances:
vessel    (C026), berth     (C068), 3         (C156) -- one        (C139), copper    (C056), vessel    (C156)
vessel    (C026), load      (C068), container (C156) -- one        (C139), nickel    (C056), vessel    (C156)
vessel    (C026), load      (C068), container (C156) -- one        (C139), oil       (C056), vessel    (C156)

x, C_53, C_183 -(10)- C_67, sell, x  Encountered 6 instances:
t-bill    (C077), fall      (C053), polish    (C183) -- ministry   (C067), sell      (C064), t-bill    (C137)
gold      (C193), ease      (C053), london    (C183) -- imf        (C067), sell      (C064), gold      (C083)

x, C_68, C_156 -(10)- C_22, expect, x  Encountered 8 instances:
vessel    (C026), berth     (C068), container (C156) -- delay      (C022), expect    (C095), vessel    (C156)
vessel    (C026), berth     (C068), 2         (C156) -- delay      (C022), expect    (C095), vessel    (C156)
cargo     (C026), bulk      (C068), loading   (C156) -- delay      (C022), expect    (C095), cargo     (C083)

x, C_40, C_146 -(10)- palestinian, C_89, x  Encountered 4 instances:
israeli   (C065), wound  (C040), palestinian (C146) -- palestinian (C074), stone    (C089), israeli   (C058)
soldier   (C167), shoot  (C040), policemen   (C146) -- palestinian (C074), pelt     (C089), soldier   (C146)
soldier   (C167), shoot  (C040), palestinian (C146) -- palestinian (C074), pelt     (C089), soldier   (C146)

x, C_95, C_27 -(10)- C_58, cash, x  Encountered 4 instances:
hog       (C091), see       (C095), steady    (C027) -- steady     (C058), cash      (C097), hog       (C196)
hog       (C091), expect    (C095), lower     (C027) -- lower      (C058), cash      (C097), hog       (C196)
hog       (C091), see       (C095), steady    (C027) -- lower      (C058), cash      (C097), hog       (C196)

x, C_31, C_146 -(10)- prosecutor, C_46, x  Encountered 5 instances:
court     (C067), sentence  (C031), people    (C146) -- prosecutor (C111), tell      (C046), court     (C102)
court     (C067), sentence  (C031), man       (C146) -- prosecutor (C111), tell      (C046), court     (C102)
court     (C067), convict   (C031), son       (C146) -- prosecutor (C111), ask       (C046), court     (C102)

x, C_74, C_90 -(10)- little, C_22, x  Encountered 17 instances:
share     (C056), end       (C074), well      (C090) -- little     (C112), run       (C022), share     (C087)
bond      (C070), close     (C074), little    (C090) -- little     (C112), change    (C022), bond      (C087)
t-bill    (C077), open      (C074), little    (C090) -- little     (C112), change    (C022), t-bill    (C137)

x, schedule, C_11 -(10)- C_80, qualify, x  Encountered 150 instances:
issue     (C139), schedule  (C017), following (C011) -- fsa        (C080), qualify   (C038), issue     (C068)
issue     (C139), schedule  (C017), insurance (C011) -- ambac      (C080), qualify   (C038), issue     (C068)
issue     (C139), schedule  (C017), insurance (C011) -- mbia       (C080), qualify   (C038), issue     (C068)

x, C_54, sep -(10)- C_64, wed, x  Encountered 20 instances:
u.k.      (C152), retail    (C054), sep       (C077) -- 0830       (C064), wed       (C034), u.k.      (C077)
u.k.      (C152), retail    (C054), sep       (C077) -- 0830       (C064), wed       (C034), u.k.      (C077)
u.k.      (C152), credit    (C054), sep       (C077) -- 0930       (C064), wed       (C034), u.k.      (C077)
```

9.5 Patterns with Object - Subject Anchor

```
C_26, C_68, x -(10)- x, C_56, C_156  Encountered 10 instances:
vessel    (C026), berth     (C068), one       (C103) -- one       (C139), copper    (C056), vessel    (C156)
cargo     (C026), bag       (C068), rice      (C196) -- rice      (C193), sugar     (C056), container (C156)
vessel    (C026), berth     (C068), container (C156) -- container (C157), sugar     (C056), waiting   (C156)
vessel    (C026), berth     (C068), one       (C103) -- one       (C139), oil       (C056), vessel    (C156)

C_157, C_68, x -(10)- x, C_56, C_156  Encountered 4 instances:
3         (C157), load      (C068), rice      (C196) -- rice      (C193), sugar     (C056), waiting   (C156)
3         (C157), discharge (C068), container (C156) -- container (C157), sugar     (C056), vessel    (C156)
loading   (C157), bulk      (C068), fertilise (C130) -- fertilise (C157), metal     (C056), vessel    (C156)

C_19, C_64, x -(10)- x, C_45, C_132  Encountered 4 instances:
rival      (C019), lease    (C064), piece     (C083) -- piece     (C188), include   (C045), device    (C132)
opposition (C019), rebuild  (C064), bridge    (C186) -- bridge    (C047), prevent   (C045), equipment (C132)

C_157, C_71, x -(10)- x, C_68, delay  Encountered 4 instances:
4         (C157), scrap     (C071), due       (C131) -- due       (C139), iron      (C068), delay     (C032)
1         (C157), map       (C071), vessel    (C156) -- vessel    (C026), berth     (C068), delay     (C032)
loading   (C157), scrap     (C071), vessel    (C156) -- vessel    (C026), berth     (C068), delay     (C032)

C_189, C_22, x -(10)- x, strike, C_138  Encountered 27 instances:
citibank  (C189), set       (C022), warrant   (C057) -- warrant   (C015), strike    (C004), price     (C138)
goldman   (C189), set       (C022), warrant   (C057) -- warrant   (C015), strike    (C004), premium   (C138)
sbc       (C189), put       (C022), warrant   (C057) -- warrant   (C015), strike    (C004), price     (C138)

C_157, scrap, x -(10)- x, C_68, C_130  Encountered 6 instances:
container (C157), scrap     (C071), cargo     (C083) -- cargo     (C026), bag       (C068), flour     (C130)
loading   (C157), scrap     (C071), vessel    (C156) -- vessel    (C026), cement    (C068), cement    (C130)
container (C157), scrap     (C071), cargo     (C083) -- cargo     (C026), bag       (C068), flour     (C130)

C_39, C_21, x -(10)- x, beat, C_155  Encountered 6 instances:
sun       (C039), report    (C021), earnings  (C168) -- earnings  (C086), beat    (C010), expectation (C155)
xerox     (C039), report    (C021), earnings  (C168) -- earnings  (C086), beat    (C010), forecast    (C155)
microsoft (C039), release   (C021), result    (C168) -- result    (C063), beat    (C010), forecast    (C155)

C_37, visit, x -(10)- x, C_46, C_63  Encountered 4 instances:
pope      (C037), visit     (C044), cuba      (C085) -- cuba      (C093), invite    (C046), pope      (C063)
jiang     (C037), visit     (C044), leader    (C165) -- leader    (C051), ask       (C046), jiang     (C063)

C_80, qualify, x -(10)- x, schedule, C_11  Encountered 374 instances:
fgic      (C080), qualify   (C038), issue     (C068) -- issue     (C139), schedule  (C017), insurance (C011)
mbia      (C080), qualify   (C038), issue     (C068) -- issue     (C139), schedule  (C017), following (C011)
fsa       (C080), qualify   (C038), issue     (C068) -- issue     (C139), schedule  (C017), following (C011)

C_80, qualify, x -(10)- x, schedule, C_58  Encountered 41 instances:
fsa       (C080), qualify   (C038), issue     (C068) -- issue     (C139), schedule  (C017), three     (C058)
mbia      (C080), qualify   (C038), issue     (C068) -- issue     (C139), schedule  (C017), two       (C058)
fgic      (C080), qualify   (C038), issue     (C068) -- issue     (C139), schedule  (C017), six       (C058)

C_64, wed, x -(10)- x, C_54, sep  Encountered 13 instances:
0930      (C064), wed       (C034), u.k.      (C077) -- u.k.      (C152), credit    (C054), sep       (C077)
0930      (C064), wed       (C034), u.k.      (C077) -- u.k.      (C152), credit    (C054), sep       (C077)
0830      (C064), wed       (C034), u.k.      (C077) -- u.k.      (C152), retail    (C054), sep       (C077)

C_189, set, x -(10)- x, control, C_163  Encountered 4 instances:
sbc       (C189), set       (C022), warrant   (C057) -- warrant   (C015), control   (C005), dem       (C163)
ubs       (C189), set       (C022), warrant   (C057) -- warrant   (C015), control   (C005), sfr       (C163)

delay, expect, x -(10)- x, C_68, C_156  Encountered 25 instances:
delay     (C022), expect    (C095), vessel    (C156) -- vessel    (C026), berth     (C068), container (C156)
delay     (C022), expect    (C095), tanker    (C171) -- tanker    (C026), berth     (C068), 2         (C156)
delay     (C022), expect    (C095), vessel    (C156) -- vessel    (C026), berth     (C068), waiting   (C156)
delay     (C022), expect    (C095), due       (C131) -- due       (C139), coal      (C068), vessel    (C156)
```

Bibliography

[1] Z. Harris (1985). Distributional Structure. In: Katz, J. J. (ed.) *The Philosophy of Linguistics.* New York: Oxford University Press. pp. 26-47.

[2] Levine (1993). *English Verb Classes and Alternations – A Preliminary Investigation*. University of Chicago Press, Chicago.

[3] W. M. Zickus (1994). A comparative analysis of Beth Levin's English verb class alternations and WordNet's senses for the verb classes HIT, TOUCH, BREAK, and CUT. In *Proceedings of The Post-Coling94 International Workshop on Directions of Lexical Research* (pp. 66-74). Beijing, China: Tsinghua University.

[4] A. Douglas, C.Berwick, F. Cho, Z. Khan, N. Nomura, A. Radhakrishnan, U. Sauerland and B. Ulicny (1994). *Verb Classes and Alterations in Bangla, German, English and Korean*. MIT AI Memo 1517.

[5] Donald Hindle (1990). Noun Classification from Predicate-Argument Structures, *Proceedings of the 28th annual meeting on Association for Computational Linguistics*, Pittsburgh, Pennsylvania Pages: 268 - 275

[6] F. Pereira, N. Tishby and L. Lee (1993). Distributional Clustering of English Words, *Proceedings of the 31st ACL*, pp 183—190.

[7] G. Grefenstette (1993). *Evaluation techniques for automatic semantic extraction: Comparing syntactic and window-based approaches*. Technical report, Department of Computer Science, University of Pittsburgh.

[8] Landauer, T.K. and Dumais, S.T. (1994). Latent semantic analysis and the measurement of knowledge. In R. M. Kaplan and J. C. Burstein (Eds) Educational Testing Service Conference on Natural Language Processing Techniques and Technology in Assessment and Education . Princeton, Educational Testing Service.

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

[10] Hearst (1992). Automatic acquisition of hyponyms from large text corpora, International Conference On Computational Linguistics Proceedings of the 14th conference on Computational linguistics - Volume 2, Pages: 539 – 545, Nantes, France

[11] L. Iwanska, N. Mata and K. Kruger (1999). Fully Automatic Acquisition of Taxonomic Knowledge from Large Corpora of Texts: Limited Syntax Knowledge Representation System Based on Natural Language, Lecture Notes In Computer Science; Vol. 1609, *Proceedings of the 11th International Symposium on Foundations of Intelligent Systems* Pages: 430 - 438

[12] M. Berland and E. Cherniak (1999). Finding Parts in Very Large Corpa, *ACL 1999.*

[13] R. Girju and D. I. Moldovan. Text mining for causal relations. In proceedings of FLAIRS conference, 2002.

[14] Patrick Pantel, Deepak Ravichandran and Eduard Hovy (2004). Towards Terascale Knowledge Acquisition, Proceedings of Conference on Computational Linguistics (COLING-04), pp. 771-777, Geneva.

[15] Markert, Katja, Natalia Modjeska and Malvina Nissim. 2003. Using the Web for Nominal Anaphora Resolution, In *EACL Workshop on the Computational Treatment of Anaphora.* Budapest, Hungary.

[16] Agirre E., Ansa O., Martínez D., E. Hovy. Enriching very large ontologies using the WWW, *Proceedings of the Ontology Learning Workshop, organized by ECAI Berlin (Germany). 2000*

[17] P. Cimiano and S. Staab (2004). Learning by googling, ACM SIGKDD Explorations Newsletter Volume 6 , Issue 2 (December 2004) Pages: 24 - 33.

[18] Akrivas, Wallace, Stamou, and Kollias (2002).Context - Sensitive Query Expansion Based on Fuzzy Clustering of Index Terms, *Proceedings of the 5th International Conference on Flexible Query Answering Systems*, Pages: 1 - 11.

[19] Mandala, Tokunaga and Tanaka (1999).Complementing WordNet with Roget's and Corpus-based Thesauri for Information Retrieval, *Proceedings of EACL '99*.

[20] D. Lin (1998). Automatic Retrieval and Clustering of Similar Words. In *Proceedings of COLING/ACL -98.* pp. 768-774. Montreal, Canada.

[21] P. Pantel and D. Lin (2002). Document Clustering with Committees, *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, Tampere, Finland. Pages: 199 – 206.

[22] S. Patwardhan (2003). *Incorporating Dictionary and Corpus Information into a Context Vector Measure of Semantic Relatedness*, a MSC thesis.

[23] P. W. Foltz, W. Kintsch, and T. K. Landauer (1998). The measurement of textual coherence with latent semantic analysis. *Discourse Processes*, 25:285—308.

[24] Petersen, W. (2002). A set-theoretical approach for the induction of inheritance hierarchies. *Electronic Notes in Theoretical Computer Science*, *51.*

[25] B. Ganter and R. Wille (1997). *Formal Concept Analysis: Mathematical Foundations*, Springer-Verlag New York, Inc. Secaucus, NJ, USA.

[26] M. Sanderson, B. Croft (1999). Deriving concept hierarchies from text. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, Berkeley, California, United States Pages: 206 - 213

[27] P. Resnik (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Montreal.

[28] C. Leacock and M. Chodorow (1998). Combining local context and WordNet similarity for word sense identification. In C. Fellbaum, editor, *WordNet: An electronic lexical database*, pages 265–283. MIT Press.

[29] G. Hirst and D. St. Onge (1998). Lexical chains as representations of context for the detection and correction of malapropisms. In C. Fellbaum, editor, *WordNet: An electronic lexical database*, pages 305–332. MIT Press.

[30] McCarthy et al (2004). Finding Predominant Senses in Untagged Text, *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*.

[31] H. Schutze (1998). Automatic word sense discrimination. *Computational Linguistics*, 24(1):97--123.

[32] P. Wiemer-Hastings, A. Graesser, and K. Wiemer-Hastings (1998). Inferring the meaning of verbs from context. In *Proceedings of the 20th Annual Conference of the Cognitive Science Society*, pp. 1142–1147, Erlbaum, Mahwah, NJ.

[33] Poesio, M., Ishikawa, T., Walde, S. and Vieira, R. (2002). Acquiring lexical knowledge for anaphora resolution. *Language resources and evaluation conference LREC 2002*, Las Palmas.

[34] R. Bekkerman, R. El-Yaniv, Y. Winter, and N. Tishby (2001). On feature distributional clustering for text categorization. In *ACM SIGIR*, pages 146-153.

[35] D. Harman (1998). Towards interactive query expansion, *Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, Grenoble, France. Pages: 321 - 331

[36] G. Miller (1990). *Wordnet: an on-line lexical database*, International Journal of Lexicography, 4(3).

[37] Karin Kipper, Hoa Trang Dang, and Martha Palmer. (2000). Class-based construction of a verb lexicon, *Proceedings of AAAI-2000*, p. 691-696.

[38] C. Baker, C. Fillmore and J. Lowe (1998). The Berkeley FrameNet Project, *Proceedings of the 17th international conference on Computational linguistics - Volume 1*, Montreal, Quebec, Canada

[39] C. J. Fillmore(1976): Frame semantics and the nature of language. In *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*, Volume 280: 20-32.

[40] P. Kingsbury, M. Palmer, M. Marcus (2002). Adding semantic annotation to the Penn TreeBank. In: *Proceedings of the Human Language Technology Conference* (HLT'02).

[41] M. Marcus, B. Santorini, and M. Marcinkiewicz (1993). Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19:313--330.

[42] T. Chklovski and P. Pantel. (2004)..VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-04)*. Barcelona, Spain.

[43] S. T. Dumais, J. Platt, D. Heckerman and M. Sahami (1998). Inductive learning algorithms and representations for text categorization. *In Proceedings of ACM-CIKM98*, pp. 148-155.

[44] M. Hearst (1994). Multi-paragraph Segmentation of Expository Text, in *Proceedings of the ACL*.

[45] D. Lin and P. Pantel (2001). DIRT---Discovery of inference rules from text. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

[46] P. Pantel and D. Ravichandran (2004). Automatically Labeling Semantic Classes, *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*

[47] G. Riccardi and S. Bangalore (1998). Automatic Acquisition of Phrase Grammars for Stochastic Language Modeling, *6th Workshop on Very Large Corpora*, pp. 186-198, Montreal.

[48] A. Maedche and S. Staab (2000). *Discovering Conceptual Relations from Text*. Technical Report 399, Institute AIFB, Karlsruhe University.

[49] R. Srikant and R. Agrawal (1995). Mining generalized association rules. In *Proc. of VLDB '95*, pages 407–419.

[50] A. Colmerauer and R. Kowalski (1972). PROLOG.

[51] R. Fikes and N. Nilsson (1971). STRIPS- A New Approach to the Application of Theorem Proving to Problem Solving, *Artificial Intelligence*, 2:189-208.

[52] Brian R. Gaines and Mildred L. G. Shaw (1993). *Eliciting Knowledge and Transferring it Effectively to a Knowledge-Based System*, Knowledge Science Institute, University of Calgary, Alberta, Canada

[53] Ido Dagan, Bernardo Magnini and Oren Glickman. 2005. The PASCAL Recognising Textual Entailment Challenge. In *Proceedings of Pascal Challenge Workshopon Recognizing Textual Entailment*.

[54] D. Lin (1994). PRINCIPAR---An Efficient, broad-coverage, principle-based parser. In *Proceedings of COLING-94.* pp.42--488, Kyoto, Japan.

[55] D. Lin (1998), Dependency-based Evaluation of MINIPAR. In *Workshop on the Evaluation of Parsing Systems*, Granada, Spain, May, 1998.

[56] Tishby, Pereira and Bialek (1999). The Information Bottleneck Method, *Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing*. p. 368-377.

[57] N. Slonim, N. Friedman, and N. Tishby (2002). Unsupervised document classification using sequential information maximization. In *Proceeding of SIGIR'02, 25th ACM International Conference on Research and Development of Information Retrieval*, Tampere, Finland. ACM Press, New York, USA.

[58] Matsuzaki, Takuya, Yusuke Miyao and Jun'ichi Tsujii. (2003). An Efficient Clustering Algorithm for Class-based Language Models. In the Proceedings of the Seventh Conference on Natural Language Learning (CoNLL) at HLT-NAACL 2003. pp. 119--126.

[59] D. Lin and P. Pantel (2001). Induction of Semantic Classes from Natural Language Text. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2001*. pp. 317-322

[60] R. Gwadera, M. Atallah and W. Szpankowski (2005). Reliable Detection of Episodes in Event Sequences, *Knowledge and Information Systems*, 7, 415 - 437; also in *Third IEEE International Conference on Data Mining*, 67-74, Florida, 2003.

[61] P. Billingsley (1986), *Probability and measure*, John Wiley, New York.

[62] Pedersen, Patwardhan, and Michelizzi (2004). WordNet::Similarity - Measuring the Relatedness of Concepts, *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*, July, 2004, San Jose, CA.