P (punctuation), **PRN** (parenthetical), **PRT** (particle), **ROOT** (root clause).The final list of dependencies is shown in Table 2.3.

Table 2.3 Syntactic Dependencies

Dependency	Description
ADV	Unclassified adv
AMOD	Modifier of adj or adv
LGS	Logical subj
NMOD	Modifier of nom
OBJ	Direct or indirect obj
PMOD	Preposition
PRD	Predicative compl
SBJ	Subject
VC	Verb chain
VMOD	Modifier of verb
empty	No dependency

Once the corpus is preprocessed, the process of matrix extraction is mostly automated. For the matrix, we have only generated vectors for words that appear at least 5 times in the corpus. Out of them we have used only the vectors of the 10,000 most frequent words for the clustering process.

For the clustering process, we configure CLUTO to use direct clustering, based on the H2 criterion function, with 25 features per cluster. We have ran the clusterization multiple times, ranging from 100 to 1,000 clusters. We then used CLUTO's H2 metric to determine the optimal number of clusters, which has been 800 for all of the experiments.

2.3.3 Grouping with Word2Vec

Word2Vec is based on the methodology proposed by Mikolov et al. [2013a]. It takes a raw corpus and a set of parameters and generates vectors and groups. The algorithm of Word2Vec is based on a two layer neural network that are trained to reconstruct linguistic context of words. Word2Vec includes two different algorithms - Continuous Bag-of-Words (CBOW) and Skip-Gram. CBOW learns representations based on the context as a whole - all of the words that co-occur with the target word in a specific window. Skip-Gram learns representation based on each single other word within a specified window. When using Word2Vec usually the emphasis is put on the choice of the paremeters for the algorithm, and not on the specifications of corpus. However, we consider that the specifications of

the corpus (size and linguistic preprocessing) can largely affect the quality of the obtained results.

By default Word2Vec works with a raw corpus. Neither of the two models makes explicit use of morpho-syntactic information. However, by modifying the corpus, some morphological information can be used implicitly. If the token is replaced by its corresponding lemma or by the lemma and part of speech tag in a "lemma_pos" format, the resulting vectors would be different: using the lemma would generate only one vector for the word as opposed to separate vector for every word form; using PoS can make a distinction between homonyms with same spelling and different PoS. As part of our work we wanted to examine how linguistic preprocessing can affect the quality of the vectors. For that reason we created three separate corpus samples - one raw corpus, one where each token was replaced by its lemma, and one where each token was replaced by "lemma_pos". We generated vectors separately for each of the corpora. Unfortunately, there was no trivial way to introduce syntactic information implicitly in the models of Word2Vec.

2.4 Experiments

In this section we present the setup for the different experiments (Section 2.4.1), the evaluation criteria (Section 2.4.2), and the obtained results (Section 2.4.3).

2.4.1 Setup

We carried out a total of 15 experiments - 3 experiments using CLUTO and 12 experiments using Word2Vec. For the experiments with CLUTO, the only variation between the experiments was the size of the corpus: 4M tokens, 20M tokens, and 40M tokens[5]. In all the experiments we used the preprocessing described at Section 2.3.2, we generated vectors for the 10,000 most frequent words and we split them into 800 clusters. For the experiments with Word2Vec, we changed three parameters of the experiments: (1) the algorithm (CBOW and Skip-Gram), (2) the linguistic preprocessing of the corpus (raw, lemma, lemma and PoS), and (3) the size of the corpus (4M, 20M, and 40M). We carried out 9 experiments with CBOW (all size and preprocessing combinations) and 3 experiments with Skip-Gram (the three variants of the 40M corpus). Mikolov et al. [2013a] identify two important parameters to be set up when using Word2Vec: the vector size and the window size. For the window size, we used 8, which is the recommended

[5]The 40M corpus contains in itself the 20M corpus. The 20M corpus contains in itself the 4M corpus. The same corpora has been used for the experiments with both CLUTO and with Word2Vec.

value. For the vector size, Mikolov et al. [2013a] show that increasing vector size from 100 to 300 leads to significant improvement of the results, however further increase does not have big impact. For that reason we have chosen vector size of 400, which is above the recommended minimum. For the number of groups we used 800: the same number that was determined optimal for CLUTO. For the number of lemmas, we used the 10,000 most frequent ones, the same setup as with CLUTO.

2.4.2 Evaluation

The two methodologies and all of the different setups are evaluated based on the quality of the obtained groups. We consider two criteria: 1) The semantic relatedness between the words in each group; and 2) The PoS coherence of the groups. The PoS coherence is a secondary criterion which should be considered in addition to the semantic relatedness. Our intuition is that groups that are semantically related and PoS coherent are a better resource than groups that are only semantically related. For evaluating the semantic relations of the words in the groups, we present two methodologies - an automated method based on WordNet distances and a manual evaluation done by experts on a subset of the groups in each experiment. The PoS coherence is calculated automatically.

There is no universal widely accepted criteria for determining the semantic relations between two words. Two of the most common approaches are calculating WordNet distances and expert intuitions. We used both when evaluating the quality of the obtained groups.

For the WordNet similarity evaluation, we use the WordNet interface built in NLTK [Bird et al., 2009]. We calculate the Leacock-Chodorow Similarity[6] between each two words[7] in every group. We then sum all the obtained scores and divide them by the number of pairs to obtain average WordNet similarity for each method.

For the expert evaluation, we selected a subset of groups, generated in each experiment[8]. Three experts were asked to rate each group on a scale from 1 (unrelated) to 4 (strongly related)[9]. We calculate the average between all of the scores

[6]It calculates word similarity, based on the shortest path that connects the senses and the maximum depth of the taxonomy in which the senses occur.

[7]The calculation is based on the first sense of every word

[8]We selected the groups based on a word they contain - three verb groups (the ones that contain "say", "see", "want"), 3 noun groups ("person", "year", "hand"), 1 adjective group("good"), 1 adverb group("well). All of the selected words are among the 100 most commonly used words of English.)

[9]In the detailed description of the scale given to the experts: 1 corresponds to "no semantic relation"; 2 corresponds to "semantic relation between some words (less than 50% of the group); 3 corresponds to "semantic relation between most of the words in the corpus (more than 50%), but