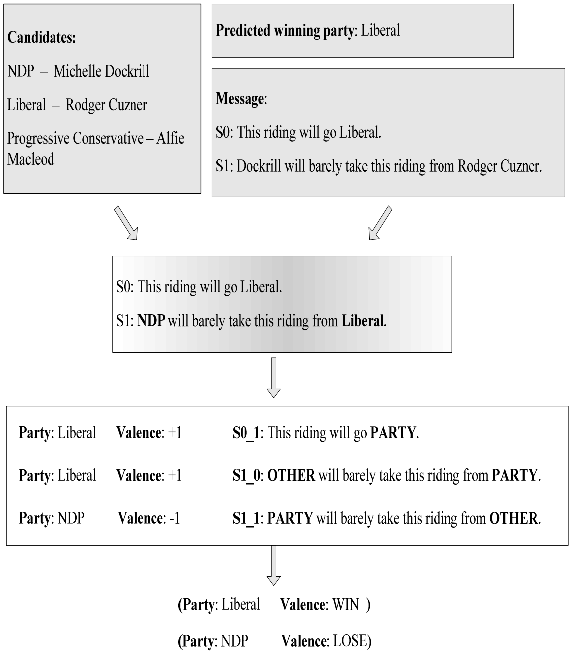

Figure 2. An example of feature generalization of a message

dicts a particular party to win, sentences which mention that party in the message also imply that it will win. Conversely all other parties are assumed to be in sentences that imply they will lose. As shown in Section 3.2, a message (*M*) in our corpus has a label of a party (P_w) that the author of *M* predicts to win. After breaking sentences in *M*, we duplicate a sentence by the number of unique parties in the sentence and modify the duplicated sentences by substituting the party names with PARTY and OTHER in order to generalize features.

Consider the following sentence:

"**Dockrill** will barely take this riding from **Rodger Cuzner**"

which gets re-written as:

"**NDP** will barely take this riding from **Liberal**"

because Dockrill is an NDP candidate and Rodger Cuzner is a Liberal candidate. Since the sentence contains two parties (i.e., NDP and Liberal), the algorithm duplicates the sentence twice, once for each party (see Lines 4–8 in Table 3)[5]. For NDP, the algorithm determines its Valence as -1 because NDP is not equal to the predicted winning party (i.e., Liberal) of the message (see Lines 4–5 in Table 3). Then it generates a generalized sentence by substituting NDP with PARTY and Liberal with OTHER (Lines 6–7). It returns (NDP, -1, "PARTY will barely take this riding from OTHER"). For Liberal, on the other hand, the algorithm determines its Valence as +1 since Liberal is the same as the predicted winning party of the message. After similar generalization, it returns (Liberal, +1, "OTHER will barely take this riding from PARTY").

Note that the final result of the feature generalization algorithm is a set of triplets: (Party, Valence, Generalized Sentence). Among a triplet, we use (Valence, Generalized Sentence) to produce feature vectors for a machine learning algorithm (see Section 4.2) and (Party, Valence) to integrate system results of each sentence for the final decision of Party and Valence of a message (see Section 4.3). Figure 2 shows an example of the algorithm.

4.2 Classification Using SVMs

In this step, we use Support Vector Machines (SVMs) to train our system using the generalized features described in Section 4.1. After we obtained examples of (Valence, Generalized Sentence) in the feature generalization step, we modeled a subtask of classifying a Generalized Sentence into Valence towards our final goal of determining (Valence, Party) of a message. This subtask is a binary classification since Valence has only 2 classes: +1 and -1[6]. Given a generalized sentence "OTHER will barely take this riding from PARTY" in Figure 2, for example, the goal of our system is to learn *WIN* valence for *PARTY*. Features for SVMs are extracted from generalized sentences. We implemented our SVM learning model using the SVM^{light} package[7].

4.3 SVM Result Integration

In this step, we combine the valence of each sentence predicted by SVMs to determine the final valence and predicted party of a message. For each party mentioned in a message, we calculate the sum of the party's valences of each sentence and

[5] In the feature generalization algorithm, we represent WIN and LOSE valence as +1 and -1.

[6] However, the final evaluation of the system and all the baselines is equally performed on the multi-classification results of messages.

[7] SVM^{light} is available from http://svmlight.joachims.org/