1 New experiments for filtering using Machine Learning for NEER

This is an erratum for the paper *NEER: An Unsupervised Method for Named Entity Evolution*, Tahmasebi et al. (2012). The machine learning filtering presented in this paper were erroneous as the set of instances also contained duplicates which resulted in an artificially high recall. In this paper, we will present new experiments for the filtering using machine learning and follow the same structure as in Tahmasebi et al. (2012).

1.1 Filtering Co-references using Machine Learning

Our third and final filtering method is based on machine learning. We use a random forest classifier (Breiman, 2001) consisting of a combination of decision trees where features are randomly selected to build each decision tree. In total, ten trees with five features each are constructed. We choose features from the similarity measures presented above. That means, for each found term–co-reference–change period tuple (w , c , p_i), we calculate the *corr*, *cov*, *rc* and *nrc* measures. We also use the average of all four measures as a fifth feature. We calculate these five measures globally and as well as locally around the change periods. For the change periods we choose one period of two years before and one period of two years after each change period which results in 15 features in total for each tuple. Finally we classify the pair as either 1, for c being a correct co-reference of w, or 0 otherwise.

1.2 Results

The results for the experiments presented next are summarized in Table 1.

NEER + Machine Learning We showed that unsupervised filtering can perform well for filtering out erroneous co-references found by NEER. In this experiment we investigated if the results could be further improved by using a limited amount of supervision for training a classifier. We used WEKA (Hall et al., 2009) and the random forest classifier. We trained our classifier on the dataset and used 10-fold stratified cross-validation to determine precision and recall. The precision presented in Table 1 is the one found by WEKA, while for recall, we use the predictions and follow the same procedure as for the other experiments.

We remove multiple instances corresponding to several change periods for one term–co-reference tuple. If a tuple (w , c , p_1) is classified as correct and a tuple (w , c , p_2) is classified as incorrect, we consider the correctly classified tuple. This because the filtering will keep at least one instance of co-reference c for the term w. As an example, in one fold we have the tuples tup1 = (*Sean Combs*, *Diddy*, 1988) and tup2 = (*Sean Combs*, *Diddy*, 2005). Tup1 is classified

Table 1: Precision and recall for the baseline and different filtering techniques.

Method	**Found periods**			**Known periods**		
	Precision (%)	Recall(%)	# co-ref	Precision (%)	Recall (%)	#co-ref
Co-occurrence	8	51	120	20	59	16
NEER	8	90	128	13	89	64
NEER + Corr	20	61	107	17	74	43
NEER + DF	33	86	28	50	81	10
NEER + ML	93	83	13	90	65	4

as incorrect by the classifier while tup2 is classified as correct. For the recall calculations, we remove tup1 from the testset and keep tup2, because *Diddy* is found as a correct co-reference for *Sean Combs* for at least one instance and thus counts should contribute to recall. If however both tup1 and tup2 where correctly classified, not removing one of the instances would have positively affected the recall.

For known bursts we got in total 3965 instances where 230 were correct co-references (we accepted combinations of correct names, e.g., *Sean Diddy Combs*). Using the classifier we were able to achieve a 90% precision and only 16 false co-references were classified as correct. The recall of the filter is 65%. For the found bursts there were 21371 instances with 587 correct co-references. The precision of 93% is comparable to that of the known bursts and only 34 false co-references were classified as true co-references. The recall is higher with 83%. The precision and recall values for the machine learning filtering is only for class 1, i.e., all instances that are classified as co-references to a term w. If we include class 0, the precision and recall is 97% for known bursts and 99% found bursts.

The large difference in recall for known and found bursts is most likely due to the small number of correct co-references for known bursts. A reason for the comparably low recall for both classes can be the acceptance of partial people names as correct. For example, for *Sean Combs* we accepted *John* to be correct because the full name is *Sean John Combs*. However, there are many *John*s and, because of the ambiguity, it is hard for the classifier to determine that *John* is a correct co-reference for *Sean Combs* based only on term frequency features.

The results show potential of the machine learning approach combined with the features chosen for the classification, in particular for the found bursts.

References

Leo Breiman. Random Forests. In *Machine Learning*, pages 5–32, 2001.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: an update. *SIGKDD Explorations*, 11(1):10–18, 2009. doi: http://doi.acm.org/10.1145/1656274.1656278.

Nina Tahmasebi, Gerhard Gossen, Nattiya Kanhabua, Helge Holzmann, and Thomas Risse. NEER: An Unsupervised Method for Named Entity Evolution Recognition. In *Proceedings of COLING 2012*, pages 2553–2568, Mumbai, India, December 2012. The COLING 2012 Organizing Committee. URL `http://www.aclweb.org/anthology/C12-1156`.