Active Learning Strategies for Support Vector Machines, Application to Temporal Relation Classification

Seyed Abolghasem Mirroshandel[⋆,†] **Gholamreza Ghassem-Sani**[⋆] **Alexis Nasr**[†]
[⋆]Computer Engineering Department, Sharif university of Technology, Tehran, Iran
[†]Laboratoire d'Informatique Fondamentale de Marseille- CNRS - UMR 6166
Université Aix-Marseille, Marseille, France
(ghasem.mirroshandel@lif.univ-mrs.fr, sani@sharif.edu, alexis.nasr@lif.univ-mrs.fr)

Abstract

Temporal relations between events is a valuable source of information which can be used in a large number of natural language processing applications such as question answering, summarization, and information extraction. Supervised temporal relation classification requires large corpora which are difficult, time consuming, and expensive to produce. Active learning strategies are well-suited to reduce this effort by efficiently selecting the most informative samples for labeling. This paper presents novel active learning strategies based on support vector machines (SVM) for temporal relation classification. A large number of empirical comparisons of different active learning algorithms and various kernel functions in SVM shows that proposed active learning strategies are effective for the given task.

1 Introduction

The identification of temporal relations between events, in texts, is a valuable information for many natural language processing (NLP) tasks, such as summarization, question answering, and information extraction. In question answering, one expects the system to answer questions such as "when an event occurred", or "what is the chronological order of some desired events". In text summarization, especially in the multi-document type, knowing the order of events is important for correctly merging related information.

Most existing algorithms for temporal relation learning are supervised, they rely on manual annotations of corpora. Producing such annotated corpora has shown to be a time consuming, hard, and expensive task (Mani et al., 2006). In this paper we explore active learning techniques as a way to control and speed up the annotation process.

In the active learning framework, the learner has control over choosing the instances that will constitute the training set. A typical active learning algorithm begins with a small number of annotated data, and selects one or more informative instances from a large set of unlabeled instances, named the pool. The chosen instance(s) are then labeled and added to other annotated data, and the model is updated with this new information. These steps are repeated until at least one termination condition is satisfied.

While there have been numerous applications of active learning to NLP researches (Settles and Craven, 2008; Xu et al., 2007), it has not been applied, to our knowledge, to temporal relation classification.

This paper presents a novel active learning strategy for SVM-based classification algorithm. The proposed algorithm considers three measures: *uncertainty*, *representativeness*, and *diversity* to select the instances that will be annotated. The method we propose is generic, it could be applied to any SVM based classification problem. Temporal relation classification has been selected, in this paper, for illustration purpose. Our experiments show that state-of-the-art results can be reproduced with a significantly smaller part of training data.

The remainder of this paper is organized as follows: Section 2 is about temporal relation classification and its related work. Section 3 describes some

Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 56–64,
Chiang Mai, Thailand, November 8 – 13, 2011. ©2011 AFNLP

of existing active learning methods. Our proposed method will be presented in Section 4. Section 5 briefly presents the characteristics of the corpora that we have used. Section 6 demonstrates the evaluation of the proposed algorithm. Finally, Section 7 concludes the paper and presents some possible future work.

2 Temporal Relation Classification with SVM

For a given ordered pair (x_1, x_2), where x_1 and x_2 are time expressions or events, a temporal information processing system identifies the type of relation that temporally links x_1 to x_2. For example in "If all the debt is converted $(event_1)$ to common, Automatic Data will issue $(event_2)$ about 3.6 million shares; last Monday $(time_1)$, the company had $(event_3)$ nearly 73 million shares outstanding.", taken from document *wsj_0541* of TimeBank (Pustejovsky et al., 2003), there are two temporal relations between pairs $(event_1, event_2)$ and $(time_1, event_3)$. The task of a temporal relation extraction system is to automatically tag these pairs with relations *BEFORE* and *INCLUDES*, respectively.

Several researchers have focused on temporal relation learning (Chklovski and Pantel, 2005; Lapata and Lascarides, 2006; Bethard et al., 2007; Chambers et al., 2007; Bethard and Martin, 2008; Mirroshandel and Ghassem-Sani, 2010; Puscasu, 2007) among which SVM has shown good performances. In this section, we describe two of the most successful SVM-based methods.

Inderjeet Mani was the first to propose an SVM-based temporal relation classification model which is based on a linear kernel (Mani et al., 2006). His system (referred to as (k_{Mani})) uses five temporal attributes that have been tagged in the standard corpora (Pustejovsky et al., 2003) plus the string of words that constitute the events, as well as their part of part of speech tags.

The other successful SVM-based temporal classification method uses a polynomial convolution tree kernel, named argument ancestor path distance kernel (AAPD), and outperforms Mani's method (Mirroshandel et al., 2010). In this model, the algorithm adds event-event syntactic properties to the simple event features described above. In order to use syntactic properties, a convolution tree kernel is applied to the parse trees of sentences containing event pairs. Through this process, useful syntactic features can be gathered for classification by SVM. The two kernels are then polynomially combined.

3 Active Learning

Supervised methods usually need a large number of annotated samples in the training phase. In most applications including temporal relation classification, the preparation of such samples is a hard, time consuming, and expensive task (Mani et al., 2006). On the other hand, all these annotated samples may not be useful, because some samples contain little (or even no) new information. Active learning algorithms overcome this problem by adding only the most informative instances labeled by an oracle (e.g., a human expert) to the learning model. Three scenarios have been proposed for the selection of instances: 1) membership query synthesis, 2) stream-based selective sampling, and 3) pool-based sampling (Settles, 2010).

In membership query synthesis, the model itself generates some instances rather than using real-world unlabeled instances (Angluin, 2004).

In stream-based selective sampling, instances are presented in a stream and the learner decides, based on its specific control measure, whether or not to query its label (Atlas et al., 1990; Cohn et al., 1994).

In pool based sampling, which is the scenario that we have chosen), a large number of unlabeled instances are collected to form the *pool* $\mathcal{U}$. The algorithm begins with a small number of labeled data $\mathcal{L}$, and then chooses one or more informative instances from $\mathcal{U}$. The chosen instance(s) are labeled and added to $\mathcal{L}$. A new model is then learned and the process iterated (Lewis and Gale, 1994).

3.1 Sample Selection Strategies

In all active learning strategies, the informativeness of each unlabeled instance is evaluated by the learner, and the most informative instance(s) are labeled. Different informativeness measures have been proposed: 1) uncertainty sampling, 2) query by committee, 3) expected model change, 4) expected error reduction, 5) variance reduction, and 6) density weighted methods (Settles, 2010).

Uncertainty sampling is the simplest and the most commonly used selection strategy. In this strategy, instances for which the prediction of the label is the most uncertain are selected by the learner (Lewis and Gale, 1994).

In query by committee, there is a committee of models trained on the current labeled data $\mathcal{L}$ based on different hypotheses. For each unlabeled instance, committee models vote for their label. The most informative instance is one with the largest disagreement on the votes (Seung et al., 1992). In the expected model change, the most informative instance is the one which causes the most change to the model (Settles et al., 2008). In expected error reduction, the learner selects instances which reduce expected error of model as much as possible (Roy and McCallum, 2001). In density weighted methods, selected instances must be both uncertain and representative in order to decrease the effect of outliers which may cause some problems especially in uncertainty sampling and query by committee strategies (Settles and Craven, 2008).

4 Proposed Algorithm

In this section, we present an active learning method based on SVM. There have been other efforts in using active learning in combination with SVM (Brinker, 2003; Xu et al., 2007), our contribution is the design of new uncertainty measures used for sample selection. In addition, the way representativeness and diversity measures are computed and combined are novel.

The algorithm is pool-based. At each iteration, k ($k \geq 1$) instances are selected from a pool $\mathcal{U}$. To select the more informative instance(s), three measures are used: *uncertainty*, *representativeness* and *diversity*. In the next subsections, we begin with an overview of multi-class classification with SVM, then introduce our three measures and describe the active learning algorithm.

4.1 Multi-class classification

In SVM binary classification, positive and negative instances are linearly partitioned by a hyper-plane (with maximum marginal distance to instances) in the original or a higher dimensional feature space. In order to classify a new instance x, its distance to the hyper-plane is computed and x is assigned to the class that corresponds to the sign of the computed distance. The distance between instance x and hyper-plane H, supported by the support vectors $x_1 \dots x_l$, is computed as follows (Han and Kamber, 2006):

$$d(x, H) = \sum_{k=1}^{l} y_k \alpha_k x_k x^T + b_0 \qquad (1)$$

where y_k is the class label of support vector x_k; α_k and b_0 are numeric parameters that are determined automatically.

For multi-class classification with m classes, in one-versus-one case, a set $\mathcal{H}$ of $\frac{m(m-1)}{2}$ hyper-planes, one for every class pair is defined. The hyper-plane that separates class i and j will be noted $H_{i,j}$. We note $\mathcal{H}_i \subset \mathcal{H}$ the set of the $m-1$ hyper-planes that separate class i from the others.

In order to classify a new instance x, its distance to each hyper-plane $H_{i,j}$ is computed and x is assigned to class i or j. At the end of this process, for every instance x, every class i has accumulated a certain number of votes, noted $V_i(x)$ (number of time a classifier has attributed the class i to instance x). The final class of x, noted $C(x)$ will be the one that has accumulated the highest number of votes.

$$C(x) = \underset{1 \leq i \leq m}{\arg\max}\, V_i(x) \qquad (2)$$

4.2 Uncertainty

Uncertainty is one of the most important measures of informativeness of an instance. If the learner is uncertain about an instance, that shows that the learning model is not able to deal with the instance properly. As a result, knowing the correct label of this uncertain instance will improve the quality of learning model.

In the process described in subsection 4.1, there are two places where uncertainty can be measured. In the first case, a decision is taken based on the difference of two distances. The smaller the difference, the less reliable the decision is. In the second case, a decision is taken based on the result of a vote. If the outcome of the vote does not show a clear majority, the decision will be less reliable.

Four measures of uncertainty are presented below, the first and second are based on distances while the third and fourth are based on the result of the vote procedure.

4.2.1 Nearest to One Hyper-Plane (NOH)

Uncertainty of an instance x is defined here as the distance to its closest class separating hyper-planes.

$$\varphi(x) = \min_{H \in \mathcal{H}_{C(x)}} |d(x, H)| \tag{3}$$

4.2.2 Nearest to All Hyper-Planes (NAH)

NAH defines the uncertainty of instance x as the sum of its distances to all its class separating hyper-planes.

$$\varphi(x) = \sum_{H \in \mathcal{H}_{C(x)}} |d(x, H)| \tag{4}$$

4.2.3 Least Votes Margin (LVM)

LVM estimates the uncertainty of an instance by the difference between the two highest votes for this instance.

$$\varphi(x) = V_i(x) - V_j(x) \tag{5}$$

where i is the class that has collected the highest number of votes and j the class that has collected the second higher number of votes.

4.2.4 Votes Entropy (VE)

VE is based on the entropy of the distribution of the vote outcome:

$$\varphi(x) = -\sum_{1 \leq i \leq m} P(V_i(x)) \log P(V_i(x)) \tag{6}$$

where $P(V_i(x))$ is simply estimated as its relative frequency $V_i(x)/m$.

4.3 Representativeness

Representativeness is another important measure for choosing samples in active learning. In figure 1, sample 1 is the nearest instance to decision boundary, it is therefore the instance that will be selected using uncertainty criterion. But it should be clear that this sample is not appropriate for selection, annotation, and addition to the training data, because it is in fact an outlier and non representative instance.

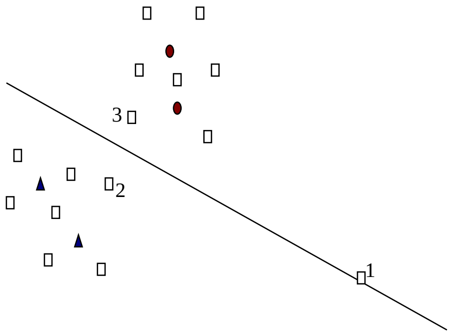

Figure 1: The weakness of uncertainty measure for dealing with outliers. Circles and triangles represent labeled instances while squares represent unlabeled instances.

This simple example shows that uncertainty measure alone is not suited to fight against outliers and noisy samples. In order to prevent the learner to select such instances, a representativeness measure ψ is used. It simply computes the average distance between an instance and all other instances in the pool:

$$\psi(x) = \frac{1}{N} \sum_{x' \in \mathcal{U}} \left| dist(x, x') \right| \tag{7}$$

where N is the number of instances in the pool, and $dist$ is the distance between two samples which can be computed by simply applying a kernel function on them:

$$dist(x_i, x_j) = kernel(x_i, x_j) \tag{8}$$

As it is shown in equation 7, the samples which are more similar to other samples of the pool will be considered to be more representative.

In order to take into account representativeness in the active learning algorithm, the distance between every sample pairs of the pool must be computed. This computation is a costly process, but these distances can be computed only once for the whole active learning algorithm. Algorithm 1 describes how representativeness and uncertainty measures have been combined.

4.4 Diversity

Diversity is the third measure that is used for instance selection. Instances that are both unreliable and representative may be very close to each other

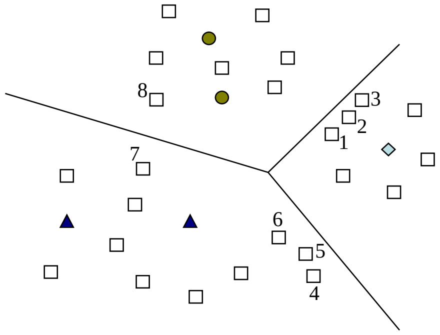

Figure 2: The necessity of applying diversity measure to select samples from the whole problem space.

and it might be interesting, in order to better cover the problem space, to select instances that are different from each other. This is done by taking diversity into account.

Figure 2 illustrates the effect of considering the diversity measure on a simple problem. In this problem, the learner chooses 4 instances for each iteration. Based on uncertainty and representativeness measures, samples 1, 2, 3, and 5 should be selected. However, 1, 2 and 3 are very similar, and only one of such samples may be enough for learning. Besides, selecting 7 and 8 will lead to a better covering of the problem space.

In our algorithm, diversity is taken into account after uncertainty and representativeness were. First, B_I instances are chosen, based on uncertainty and representativeness. A distance matrix is then constructed, based on the distance measure of equation 8. The B_I instances are then grouped into B_F ($B_F < B_I$) clusters, using hierarchical clustering and the centroid of each cluster is selected for labeling. This process is explained in algorithm 1.

4.5 Proposed Algorithm

The pseudo-code of our active learning algorithm is shown in Algorithm 1. This algorithm first trains the model based on the initial labeled data, and applies a combination of uncertainty and representativeness measures to select B_I samples from the pool. Then hierarchical clustering is applied to the extracted samples to select B_F most diverse samples. Chosen samples are then labeled and added to the training labeled set. This process is iterated until

Algorithm 1 THE PROPOSED ACTIVE LEARNING

```
$\alpha$: Uncertainty coefficient
$\mathcal{L}$: Labeled set
$\mathcal{U}$: Unlabeled pool
$\varphi(x)$: Uncertainty measure
$\psi(x)$: Representativeness measure
$B_I$: Initial query batch size
$B_F$: Final query batch size
  while termination condition is not satisfied do
    $\theta = train(\mathcal{L})$; $\mathcal{T}_I = \emptyset$;
    for $i = 1$ to $B_I$ do
      // Find most uncertain and representative instance
      $\hat{x} = \arg\max_{x \in \mathcal{U}} [\alpha\varphi(x) + (1 - \alpha)\psi(x)]$;
      $\mathcal{T}_I = \mathcal{T}_I \cup \{\hat{x}\}$;
    end for
    Apply Hierarchical clustering on $\mathcal{T}_I$ to extract set $\mathcal{T}_F$ of $B_F$ diverse samples;
    $\mathcal{U} = \mathcal{U} - \mathcal{T}_F$;
    $\mathcal{L} = \mathcal{L} \cup \mathcal{T}_F$;
  end while
```

at least one termination condition is satisfied. In our experiments, the algorithm stops when all instances of the pool were selected and labeled.

Our algorithm may seem much more costly than the original SVM algorithm. However, it is easy to show, similar to (Brinker, 2003), that it only multiply by a coefficient of N/B_F (N is the final number of labeled instances) the total computational complexity of original SVM.

5 Corpus Description

Two standard corpora were used for our expriments: TimeBank (v. 1.2)(Pustejovsky et al., 2003) and Opinion (`www.timeml.org`). TimeBank is composed of 183 newswire documents and 64 077 words, and Opinion comprises 73 documents with 38 709 words. These two datasets have been annotated with TimeML (Pustejovsky et al., 2004). There are 14 temporal relation types (*SIMULTANEOUS*, *IDENTITY*, *BEFORE*, *AFTER*, *IBEFORE*, *IAFTER*, *INCLUDES*, *IS_INCLUDED*, *DURING*, *DURING_INV*, *BEGINS*, *BEGUN_BY*, *ENDS*, *ENDED_BY*) in the TLink class of TimeML. Similar to (Mani et al., 2006; Chambers et al., 2007), we used a normalized version of these 14 temporal relation types, which contains only the following six temporal relations:

SIMULTANEOUS *ENDS* *BEGINS*
BEFORE *IBEFORE* *INCLUDES*

In order to convert 14 relations into 6, the inverse relations were omitted (e.g., *BEFORE* and *AFTER*), and *IDENTITY* and *SIMULTAENOUS*, as well as *IS_INCLUDED* and *DURING* were collapsed, respectively.

Relation Type	**OTC**
IBEFORE	131
BEGINS	160
ENDS	208
SIMULTANEOUS	1528
INCLUDES	950
BEFORE	3170
TOTAL	6147

Table 1: The normalized TLink class distribution in OTC.

In our experiments, as in several previous work, we merged the two datasets to generate a single corpus called OTC. Table 1 shows the normalized TLink class distribution (only for Event-Event relations) for OTC corpora.

6 Experimental Results

The algorithm described above was evaluated on OTC corpus with our four uncertainty measures with and without representativeness and diversity. We used random instance selection (i.e., passive learning) as the baseline strategy.

Several kernels can be used for such experiments. As explained in section 2, we decided to use the kernel proposed in (Mani et al., 2006), which we will refer to as Mani's kernel, and the Argument Ancestor Path Distance (AAPD) polynomial kernel proposed in (Mirroshandel et al., 2010). AAPD polynomial is the state of the art pattern-based algorithm, which exclusively combines gold standard features of events and grammatical structures of sentences.

All evaluations are based on a 5-fold cross validation. The original corpora was randomly partitioned into 5 parts, out of which, a single part was retained for testing the model, and the remaining 4 parts were used for the training and applying our instance selection strategies. This process was then repeated 5 times (the folds), with each of the 5 parts being used exactly once as the test data. To perform the experiments, we started from initial labeled set with 100 randomly selected samples, and in each iteration, 25 samples were selected, labeled, and added to the previously labeled set.

6.1 Uncertainty Measure Alone

Figures 3 and 4 show the result of applying our four uncertainty measures for "instance selection" in OTC, using Mani's (Figure 3) and AAPD kernels (Figure 4).

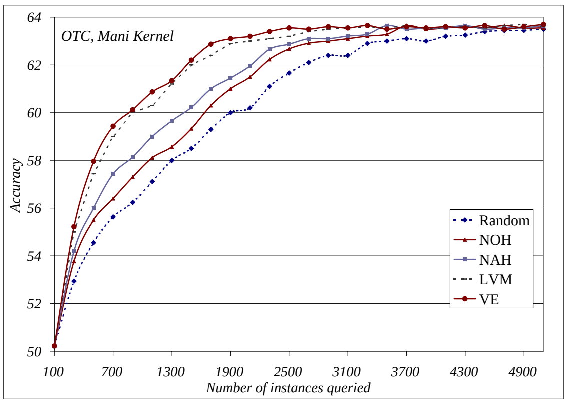

Figure 3: Learning curves for different uncertainty instance selection strategies applied to OTC using Mani's kernel.

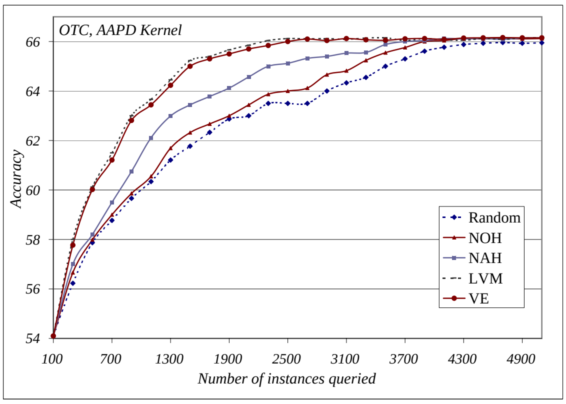

Figure 4: Learning curves for different uncertainty instance selection strategies applied to OTC using AAPD kernel.

The figures show that all proposed uncertainty instance selection strategies are effective and lead to learning curves that are above the baseline. Vote based measures have outperformed distance based ones. Among the two distance based measures, NAH led to better results than NOH, showing that averaging (aggregation) over the distances to the different separating hyperplanes is more robust than taking into account only the distance to the closest one.

The two vote based methods led to very close results, which seems to indicate that the system usually hesitates between two classes (and not more) when trying to classify an instance.

6.2 Combining Uncertainty and Representativeness Measures

Representativeness has been introduced in order to fight against outliers. Such outliers have two different origins. The first one is data sparseness: some temporal relation events are poorly represented in the data. Eliminating such instances will degrade the results on the corresponding class but will introduce less noise in the data. The second origin of outliers is the difficulty of problem, even for human annotators (Pustejovsky et al., 2003). This causes some mistakes in annotation and generates some outliers.

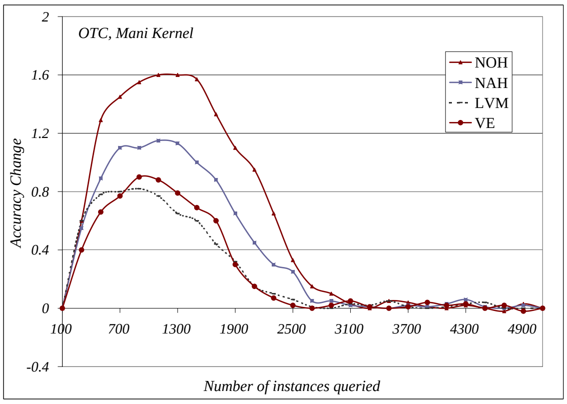

Figure 5: Accuracy improvement when adding representativeness measure to the uncertainty instance selection in Mani's kernel.

In the second series of experiments, we combined a representativeness measure with different uncertainty instance selection strategies to tackle outliers' side effects. In our different experiments, the best value for uncertainty coefficient (α) was 0.65. Figure 5 (resp. 6) shows the accuracy improvement when adding representativeness to uncertainty with Mani's (resp. AAPD) kernel. We have chosen to represent just the improvement rather than the learning accuracy, because the learning curves were not easy to compare.

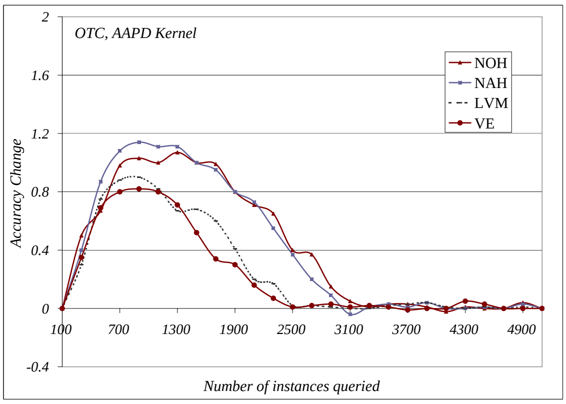

Figure 6: Accuracy improvement when adding representativeness measure to the uncertainty instance selection in AAPD kernel.

The results show that distance based measures are more sensitive to outliers than vote based ones. Figures 5 and 6 also show that the representativeness measure has less impact on AAPD kernel than it has on Mani's kernel. This is because AAPD kernel is more resistant to outliers than Mani's kernel.

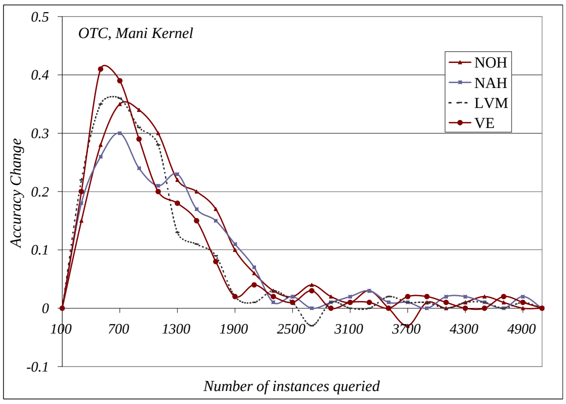

Figure 7: Accuracy improvement when adding diversity in the instance selection with Mani's kernel.

6.3 Combining Uncertainty, Representativeness and Diversity

In the last series of experiments, diversity was added to the instance selection procedure. In each iteration, first 80 instances of the pool were selected by combination of uncertainty and representativeness mea-

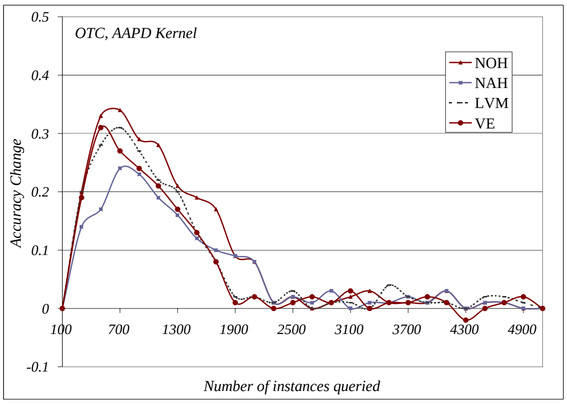

Figure 8: Accuracy improvement when adding diversity in the instance selection with AAPD kernel.

sures. Next, a hierarchical clustering method was used to select the final 25 instances. The accuracy improvement, as it is shown in Figures 7 and 8, is moderate.

The reasons why introducing diversity did not have a greater impact on the results is not clear. That may be due to the way diversity was introduced in our model. It could also come from the distribution of the data: if instances that are both unreliable and representative are not close to each other, selecting instances that are different from each other for better coverage of the problem space is not an issue. More work has to be done to investigate that point.

The final learning curves, when uncertainty, representativeness, and diversity were all considered, are shown in figures 9 and 10. As shown, vote-based uncertainty measures still obtain better results than distance based measures.

7 Conclusion

In this paper, we have addressed the problem of active learning based on support vector machines for temporal relation classification. Three different kind of measures have been used for selecting the most informative instances: uncertainty, representativeness and diversity. The results showed that the three measures improved the learning curve although diversity had a moderate effect.

Future work will focus on three points, the first one is trying other sample selection strategies, as query by committee, the second will focus on combining the two families of uncertainty measures that we have proposed: distance based and vote based. The third one is about diversity. As mentioned above, we do not know if this phenomenon is not well handeled by the model or if it is not an issue for the problem at hand.

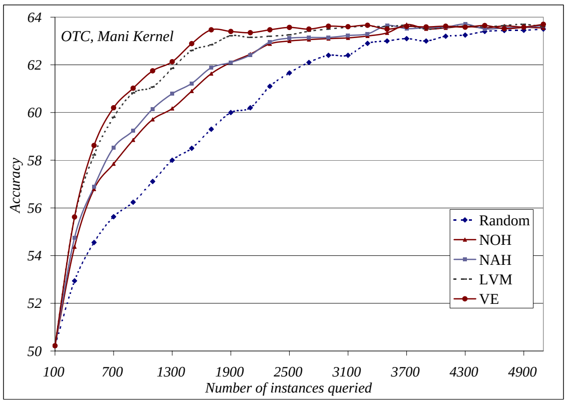

Figure 9: Learning curves for combined uncertainty, representative and diversity measures with Mani's kernel.

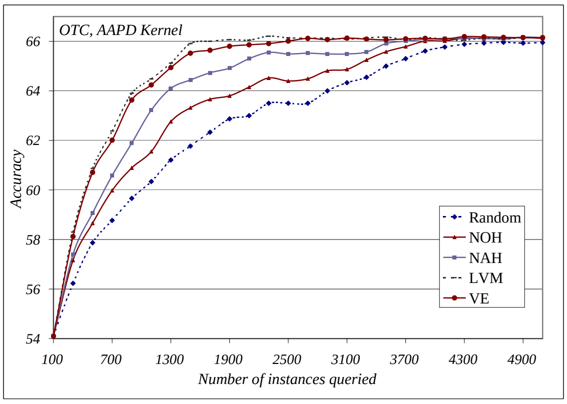

Figure 10: Learning curves for combined uncertainty, representative and diversity measures with AAPD kernel.

Acknowledgement

This work has been funded by the French Agence Nationale pour la Recherche, through the project SEQUOIA (ANR-08-EMER-013).

References

D. Angluin. 2004. Queries revisited. *Theoretical Computer Science*, 313(2):175–194.

L. Atlas, D. Cohn, R. Ladner, MA El-Sharkawi, and II Marks. 1990. Training connectionist networks with queries and selective sampling. In *Advances in neural information processing systems 2*, pages 566–573. Morgan Kaufmann Publishers Inc.

S. Bethard and J.H. Martin. 2008. Learning semantic links from a corpus of parallel temporal and causal relations. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 177–180. Association for Computational Linguistics.

S. Bethard, J.H. Martin, and S. Klingenstein. 2007. Finding temporal structure in text: Machine learning of syntactic temporal relations. *International Journal of Semantic Computing*, 1(4).

K. Brinker. 2003. Incorporating diversity in active learning with support vector machines. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, volume 20, pages 59–66.

N. Chambers, S. Wang, and D. Jurafsky. 2007. Classifying temporal relations between events. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 173–176. Association for Computational Linguistics.

T. Chklovski and P. Pantel. 2005. Global path-based refinement of noisy graphs applied to verb semantics. *Natural Language Processing–IJCNLP 2005*, pages 792–803.

D. Cohn, L. Atlas, and R. Ladner. 1994. Improving generalization with active learning. *Machine Learning*, 15(2):201–221.

J. Han and M. Kamber. 2006. *Data mining: concepts and techniques*. Morgan Kaufmann.

M. Lapata and A. Lascarides. 2006. Learning sentence-internal temporal relations. *Journal of Artificial Intelligence Research*, 27(1):85–117.

D.D. Lewis and W.A. Gale. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc.

I. Mani, M. Verhagen, B. Wellner, C.M. Lee, and J. Pustejovsky. 2006. Machine learning of temporal relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 753–760. Association for Computational Linguistics.

S.A. Mirroshandel and G. Ghassem-Sani. 2010. Temporal Relations Learning with a Bootstrapped Cross-document Classifier. In *Proceeding of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, pages 829–834. IOS Press.

S.A. Mirroshandel, G. Ghassem-Sani, and M. Khayyamian. 2010. Using Syntactic-Based Kernels for Classifying Temporal Relations. *Journal of Computer Science and Technology*, 26(1):68–80.

G. Puscasu. 2007. Wvali: Temporal relation identification by syntactico-semantic analysis. In *Proceedings of the 4th International Workshop on SemEval*, pages 484–487.

J. Pustejovsky, P. Hanks, R. Sauri, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro, et al. 2003. The timebank corpus. In *Corpus Linguistics*, volume 2003, page 40.

J. Pustejovsky, B. Ingria, R. Sauri, J. Castano, J. Littman, R. Gaizauskas, A. Setzer, G. Katz, and I. Mani. 2004. The specification language TimeML. *The Language of Time: A Reader. Oxford University Press, Oxford*.

N. Roy and A. McCallum. 2001. Toward optimal active learning through sampling estimation of error reduction. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pages 441–448. Citeseer.

B. Settles and M. Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1070–1079. Association for Computational Linguistics.

B. Settles, M. Craven, and S. Ray. 2008. Multiple-instance active learning. In *In Advances in Neural Information Processing Systems (NIPS*. Citeseer.

Burr Settles. 2010. Active Learning Literature Survey. Technical Report Technical Report 1648, University of Wisconsin-Madison.

H.S. Seung, M. Opper, and H. Sompolinsky. 1992. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294. ACM.

Z. Xu, R. Akella, and Y. Zhang. 2007. Incorporating diversity and density in active learning for relevance feedback. *Advances in Information Retrieval*, pages 246–257.