Frame Assignment with Active Learning

By:
Masood Ghayoomi

Supervisors:
Prof. Dr. Manfred Pinkal
Dr. Caroline Sporleder

Master of Science Thesis
Department of Computational Linguistics and Phonetics
Saarland University

Abstract

Recently natural language understanding is given a special attention, since in natural language processing techniques syntactic analysis such as part-of-speech tagging and parsing had a great progress and semantic analysis did not have such a rapid progress. In information extraction and question-answering systems semantic understanding techniques are required. Frame semantics structure analysis is one of the understanding techniques. In this type of analysis, the semantic roles of elements participated in the action would be identified. To determine the roles automatically, two steps are required: one is frame assignment, and the other one is role assignment. What we aim to do is assigning frames with a supervised machine learning method called 'active learning'. Supervised learning method requires a huge amount of labeled data. The aim of active learning promises to maximize the performance by minimizing the human's effort to label the data.

To our end, we have selected pool-based active learning with uncertainty sampling method; and also we have chosen 14 frequent targets from FrameNet data set for our task. Random sampling which represents the distribution of frames in the corpus would be our baseline to find how effective active learning is. Since for each target there was at least one dominant frame, we faced the imbalanced problem which might have a negative impact on the classifier; so over-sampling is used to resolve this problem.

According to the results, active learning worked out for most of the targets; for some of them it was not that much effective; and for some it had a negative impact which shows active learning could not always be a help. We have discussed this issue in details.

As we know active learning is an iterative process, it should be stopped when the classifier has reached to the maximum performance. Reaching this point is so difficult; so, we have proposed a new stopping criterion which stops active learning in a near-optimum point. This stopping criterion is based on the confidence score of the classifier on the extra unlabeled data such that it uses the variance of classifier's confidence score for a certain number of samples which are selected in each iteration.

Dedication

To my parents

and

my wife, Saeedeh

for their true love and support.

Declaration

I hereby declare that this thesis is by my own work and effort. Where other sources of information have been used, they have been acknowledged.

Saarbrücken, September 22, 2009

Masood Ghayoomi

Contents

List of Tables

List of Figures

Chapter 1
Introduction

1.1 Introduction

Recently a special attention is given to natural language understanding due to the rapid advances in sub-tasks of Natural Language Processing (NLP) techniques such as part-of-speech (POS) tagging and parsing. Considering only such syntactic analysis is not enough in some NLP applications, such as information extraction and question answering systems; and a deeper understanding is required. To achieve this goal, semantic understanding systems are needed. Frame Semantics structure analysis is one of the understanding techniques which describes abstract actions along with their participants. Considering the following example from Berkeley FrameNet project (Baker et al., 1998), the frame STATEMENT represented by the lexical unit 'said' contains elements like SPEAKER, ADDRESSEE, and MESSAGE as semantic roles to the participants of the action:

[*Message* 'I'll come to your office at 8 o'clock'] [*Speaker* Susan] **said** [*Addressee* to Kim].

This type of annotation of data to identify the semantic roles is so important in applications like information extraction, statistical machine translation, automatic

text summarization, and text data mining (Hearst, 1999); moreover such annotation might be used in question-answering systems, semantic dialogue systems, word sense disambiguation, and probabilistic language models for speech recognition systems (Gildea and Jurafsky, 2002) to have better language models.

As represented in the example, to understand a sentence, besides considering the lexical units and their syntactic relations, it is important to identify the frame assigned to the predicate and also recognize the conceptual relations between the elements of the sentence with respect to this frame.

What we aim to study is finding and assigning the high probable frames of the predicate 'verb' with the help of machine learning techniques. Supervised learning methods, as one of the machine learning techniques, require a huge amount of labeled data; so it is very difficult, time consuming, error prone, costly, and expensive to obtain the labels of the data. The supervised machine learning method that we would like to use is called 'active learning'. Active learning promises to reduce the annotation cost by decreasing the number of labeled items which are annotated by an oracle, a human annotator, and needed for the language technology; i.e. having used active learning causes to reduce the number of labeled examples needed to achieve the same level of performance on labeling all available data.

There are large bodies of active learning works in the literature, but less within NLP applications. What we aim to do is assigning frames to the 'verbs' of sentences, as our target words, with the help of active learning; and we hypothesize that using active learning for frame assignment would make a decrease on the amount of required annotated data to achieve the same level of performance. In other words, an effective active learner should achieve the highest performance for assigning frames with the lower labeled data than the baseline.

This hypothesis could be positive or negative. If it is positive, then we should find out that having a small set of labeled data selected intelligently under the shadow of active learning will have the higher or probably the same achievements compared to using a large set of labeled data selected randomly. If we consider the hypothesis

negative, then there is no effect on the amount of labeling data; so that all the data should be annotated.

The contribution behind our study is that, as we will see more on next sections, frame assignment plays a significant role in semantic role labeling and to our knowledge no previous study has been done to use active learning for frame assignment. In our research we aim to bridge the gap.

1.2 Outline of the Study

The contents of this thesis are represented as followings:

Section 2: we will have a brief glance on frame semantics and frame assignment.

Section 3: it presents a quick review on machine learning and a short description on active learning scenarios and query methods which are used in our study. Moreover, the previous studies done with active learning on NLP would be represented, and we will concentrate more on Word Sense Disambiguation (WSD).

Section 4: we will talk about mainly on collection of the data required for our task, preparing the data, and distributing the data.

Section 5: it introduces our approach in the thesis in which we will mainly present our proposed model, its components, and the toolkit used for data processing.

Section 6: we will analyze the results we have achieved after the data processing, and have discussions on our findings.

Section 7: it represents the conclusion and the summary of the work along with the future work.

1.3 Summary

In this chapter an introduction to our study was presented. As described, we are interested to do a semantic analysis on the natural language which is frame assignment with the help of a supervised machine learning method called active learning in frame semantics framework. This chapter is finished with the outline of our study.

Chapter 2

Semantic Processing

2.1 Introduction

During the last decade, syntactic analysis had a great progress and success in POS tagging and parsing with a high percentage of accuracy on correct analysis as various taggers and parsers are available nowadays. However, semantic processing did not have such a successful background. Since in many recent NLP applications such as information extraction, question answering systems, dialogue systems, and machine translation the semantic representation of a text is required, as a result it highlights the importance of semantic analysis. To this aim, we focus on frame semantic structure analysis as a framework for semantic representation to provide the knowledge about the actions, the participants of the action, and the relations between them. In the following subsections, frame semantics and shallow semantic processing would be discussed.

2.2 Frame Semantics

Frame semantic is a framework to describe the meaning representation in an abstract level such that the actions along with their participants are realized. One of the origins of frame semantics is the case grammar of Fillmore (1968) known as ‘case

frame'. In Fillmore's view, a frame is considered as an abstract scene (a schematized scene) having some participants as the arguments of this predicate, and some sentences to describe the scene. In fact frames are the conceptual structures for the background knowledge of the abstract scenes represented by lexical units and provide context to the elements of the action. Consequently, in order 'to understand the semantic structure of the predicate [e.g. the verb] it is necessary to understand the properties of such schematized scene' (Fillmore, 1982: 115); so that we could realize the semantic roles of each participant of the scene with respect to the characterization of the predicate. Fillmore (1985) considers the notion of frame for 'semantic understanding' to determine what situation a sentence fits for the hearer to provide the interpretation of a sentence. In Fillmore's recent view, the semantic understanding is compositional such that knowledge of words, phrases and the grammatical constructions for an interpretation are taken into consideration.

Based on what is said, frames have two levels: one is the conceptual level in which a frame models a specific schematized situation and represents its relevant background knowledge; the other one is the linguistic level in which a frame is considered as a semantic class and contains all predicates which are capable of expressing the target situation (Padó, 2007). On the conceptual level, abstract roles named 'semantic roles' are assigned to the participants of the action with respect to the predicate. Such information is different from the syntactic functions of the arguments (e.g. subject, object) from syntactic point of view. Fillmore (1968; 1971) introduced a set of semantic roles such as AGENTIVE, EXPERIENCER, INSTRUMENTAL, DATIVE, FACTITIVE, SOURCE, GOAL, LOCATIVE, TIME, PATH, and OBJECTIVE. These semantic roles are in fact the 'frame elements' (FE) of a relevant frame which is evoked by some specific predicates called 'frame evoking elements' (FEE).

Among the applications of frame semantics, its usage in lexicography is the most important one to impose a certain classification scheme on the lexicon (Fillmore, 1994). The Berkeley FrameNet project (Baker et al., 1998) which has compiled a semantic lexicon for English based on frame semantics is the other application.

FrameNet focuses on four lexical categories namely verbs, nouns, adjectives, and prepositions. Moreover, multiword expressions, hyphenated words, and idiomatic expressions are also defined as a lexical unit in which the appropriate frames are assigned to them without analyzing their internal structures (Ruppenhofer et al., 2006).

Each frame contains 5 types of information: presenting the frame name as the semantic class; defining the frame in natural language; listing the frame elements involved in the situations and presuppositions of the situation; presenting FEEs as lemma-dot-POS; and providing some examples as annotated corpus.

In Table 1 the frame COMMITMENT could be found in which based on the given definition, the frame elements are listed and evoking elements which share the same properties are presented beside the annotated examples:

Table 1. A sample frame of FrameNet for the frame COMMITMENT

Frame: COMMITMENT	
Def.	A Speaker makes a commitment to an Addressee to carry out some future action. This may be an action desirable (as with promise) or not desirable (as with threaten) to the Addressee.
FEs	Speaker The Speaker is the person who commits him/herself to do something Addressee The Speaker's commitment can be made to an Addressee. Message An expression of the commitment made by the Speaker Topic The topic about which the Speaker makes a promise. Medium The Medium is the physical entity or channel used to transmit the Message.
FEEs	consent.v, covenant.n, covenant.v, oath.n, vow.n, pledge.n, pledge.v, promise.n, promise.v, swear.v, threat.n, threaten.v, undertake.v
Ann.	[*Speaker* Democratic audiences] had to **consent** [*Message* to this approach]. [*Speaker* The politicians] made vague **promises** [*Topic* about independence]. [*Message* 'I'll be back , '][*Speaker* he] **threatened**.

FrameNet has another property too, that is having relations between two frames. The frame which is more abstract and less independent would be called 'super-frame' and the other frames which are less abstract and more dependent are called 'sub-frames'. There are 8 types of relations between sub-frames and super-frames as given in Tabel 2 (Ruppenhofer et al., 2006):

Table 2. Kinds of relation between sub-frames and super-frames in FrameNet

Relation	Sub-frame	Super-frame
Inheritance	Child	Parent
Perspective-on	Perspectivized	Neutral
Subframe	Component	Complex
Precedes	Later	Earlier
Inchoative-of	Inchoative	State
Causative-of	Causative	Inchoative/State
Using	Child	Parent
See-also	Referring Entry	Main Entry

Among the relations, we only describe the inheritance relation as a sample. Having some properties in common for different frames, makes it possible to do generalization and have a more general frame as the 'domain' of the frames and represent an inheritance hierarchical structure between the sub-frames as children and super-frames as parents. It should be added that the FEs of the parents are bounded to the FEs of the children. As an example, the CONVERSATION, QUESTIONING, and COMMITMENT frames _which are sub-frames_ with their individual relevant FEs and FEEs are under the domain COMMUNICATION _which is the super-frame_ represented in Figure 1:

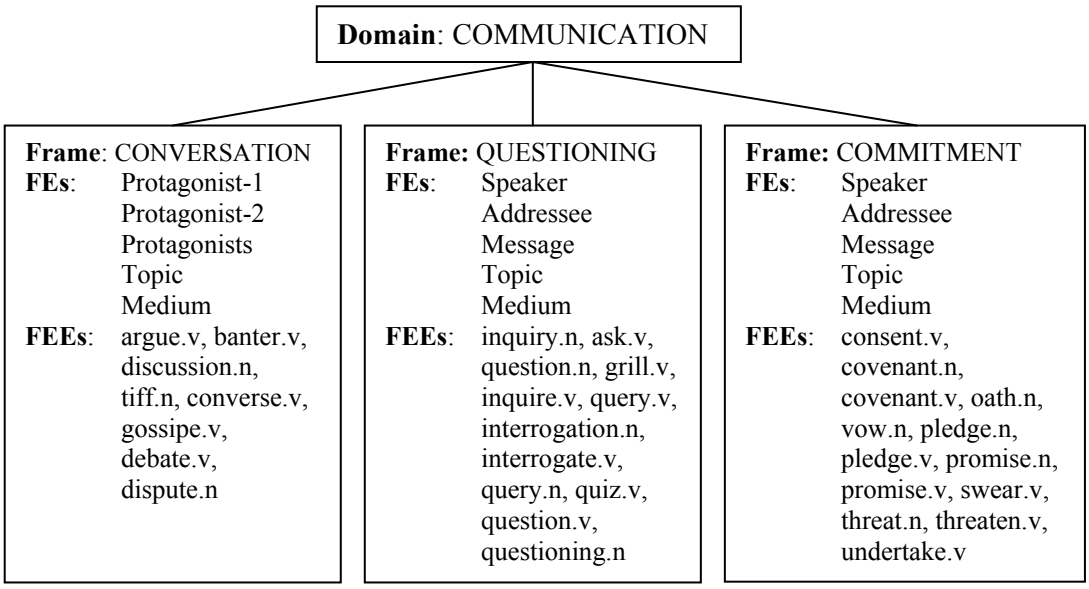

Figure 1. Inheritance relation on FrameNet

Now that we are familiar with frame semantics, how could we benefit of it in computational linguistics?

2.3 Shallow Semantic Parsing

Nowadays we have access to a huge amount of linguistic data that are freely available from online sources, so that if we do the frame semantic analysis automatically then we are doing 'shallow semantic parsing'. In such processing, we try to assign frames and semantic roles to free texts. Although in theoretical linguistics there has been a lot of preliminary research on semantic role labeling, in computational linguistics it is just started by Gildea and Jurafsky (2002), and continued at CoNLL (Carreras and Márquez, 2004; 2005) and SENSEVAL-3 (Mihalcea and Edmonds, 2004).

Thomson et al. (2003) has done a research on semantic role labeling in which the appropriate frames and the relevant semantic roles of the frames are assigned in a single step. In contrast, Erk and Padó (2006) have considered shallow semantic parsing in two dependent steps: the first step is identifying the frame which is evoked by the predicate to determine the unique frame that is appropriate for the sample. This step is called 'frame assignment'. Erk (2005) has considered this step as a disambiguation step to assign the highest probable frame to the predicate. The next step is realizing the arguments of the predicate and defining the frame elements (semantic roles) to the constituental arguments with respect to the given frame. This step is called 'role assignment'. Believing to have a two-step processing in shallow semantic parsing, frame assignment plays a significant role as the preceding step to role assignment, since the set of appropriate frame elements is dependent on the assigned frame.

In Figure 2 a snapshot for frame and role assignments is presented which is the visualized output of Shalmanser (SHALlow seMANtic parSER; Erk and Padó, 2006) by Salto tool (Burchardt et al., 2006). The semantic frame is represented in the dark gray rectangles which are APPEARANCE and STATEMENT for the evoking

elements ‘appeared’ and ‘spoke’ respectively. The white rectangles present the relevant semantic roles of each frame which are PHENOMENIN and PERCEIVER-PASSIVE for the former frame; and SPEAKER, ADDRESSEE, and TOPIC for the latter frame:

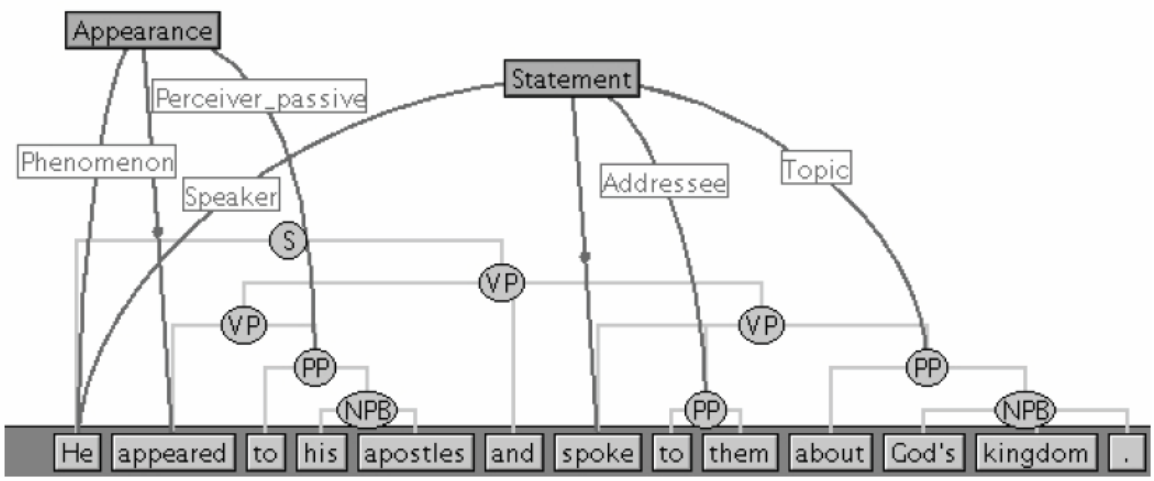

Figure 2. A snapshot for frame and role assignments

Since shallow semantic parsing relies heavily on the use of supervised methods, large annotated corpora as training data are required. Since there is a huge amount of free data available and labeling such data is time consuming and costly, then we should use machine learning techniques to minimize human’s effort to label the data. In the next section, we will have a glance on our target machine learning methods which is active learning.

2.4 Summary

In this chapter we briefly introduced frame semantic structure analysis. Then we described a bit about FrameNet as a framework for frame semantics. Talking about FrameNet properties and applications, we talked about shallow semantic processing and its steps. The primer step was frame assignment which is the focus of this research.

Chapter 3

Active Learning Method

3.1 Introduction

Machine Learning is one of the sub-branches of Artificial Intelligence in which tries to simulate the intelligent abilities of a human being in machines. Learning in machine learning is an 'inductive inference' from the seen data in which the system is trained with and trying to make predictions on unseen events. The learning could be unsupervised and supervised. In former, the learner tries to uncover the hidden regularities in the data as the learner clusters the unlabeled data based on their similarities; while in latter, the labels are already given to the data and the learner tries to recognize the patterns of the data and classify them based on the initial data that the learner is trained with (Tong, 2001; Rätsch, 2004; Settles, 2009).

'Active learning' is one of the supervised machine learning methods (Settles, 2009) to create labeled data with the help of human annotations in a feedback loop fashion (Busser and Morante, 2005). This learning method is in contrast to 'passive learning' in which all samples are annotated with the oracle and the system is trained with such data (Tong, 2001).

Since in supervised learning labeled data is required, so the labeling task is costly, expensive, time consuming, and error prone to obtain. The motivation behind active learning is maximizing the performance by minimizing the human's effort as much as possible for labeling the data (Thompson et al., 1999). Another property of active learning is that it is an 'iterative' process (Busser and Morante, 2005). In each iteration, firstly the learner is trained with the training data; then a small subset of the unlabeled data is selected and handed to the oracle to label them; and finally the newly labeled data would be added to the initial learner's training set and the learner will be retrained. This process continues in a loop till it terminates.

3.2 Active Learning Scenarios

Looking at the previous works in the literature (Cohn et al., 1994; Lewis and Gale, 1994; Dagan and Engelson, 1995; Freund et al., 1997; Fujii et al., 1998; McCallum and Nigam, 1998; Thompson et al., 1999; Schohn and Cohn, 2000; Roy and McCallum, 2001; Tong, 2001; Tong and Koller, 2001; Baram et al., 2004; Chen et al., 2006; Hoi et al., 2006), there are two major learning scenarios which are so popular among researches and frequently used in various NLP tasks as we will see below. The two learning scenarios are stream-based sampling and pool-based sampling.

3.2.1 Stream-based sampling

In 'stream-based sampling' scenario, the learner takes one sample at a time from the distributed data and tries to makes a decision on this sample whether to select and hand it out to the oracle for labeling or disregard it.

3.2.2 Pool-based sampling

In 'pool-based sampling' scenario, at first the learner takes all the samples and rank them based on the classifier's prediction ascendingly. Then, the learner selects the

top *k* samples from this ranked list on each iteration and hand them out one by one to the oracle for labeling.

In pool-based scenario the pool of the unlabeled data is fixed and not changed; while stream-based scenario is the online version of the pool-based as it immediately asks the oracle for a label after it has been classified (Baram et al., 2004). We can conclude that stream-based is differentiated from pool-based scenario with respect to whether in the model the selected samples are immediately handed to the oracle for labeling or at first all samples are ranked and from this ranked data samples are selected to be labeled.

3.3 Query Strategies in Active Learning

The samples that are selected should be hard and very informative to be effective for the task. To do the sample selection, there are some query methods which are independent to the active learning scenarios introduced above.

Among the various query methods for active learning used in different applications, Settles (2008; 2009) has presented the most complete and comprehensive classification for the various query methods in active learning. Among them only two of them are described briefly here which are widespread and frequently used in different tasks including NLP.

3.3.1 Uncertainty Sampling

The most well-known and simple sample selection method is ‘uncertainty sampling’ introduced by Lewis and Gale (1994). In this query method, active learner hands out the samples to the oracle which are the most uncertain. Considering the prediction of the classifier for unlabeled data, we will have a border between ‘uninformative’ and ‘informative’ samples. Uninformative samples are the ones which have the highest prediction certainty for assigning labels; so, this kind of samples are not very useful for active learner and it is not necessary to ask the oracle to label them. The

informative samples are the ones that the classifier does not have a good prediction to assign a label with high confidence. Such data that have lower certainty are interesting for active learner. The border line between the informative and uninformative samples could be defined with a score as the confidence of the classifier for the label it assigns to. In fact this 'confidence score' is the prediction of the classifier with the highest probability for the label of the sample (Busser and Morante, 2005). As it could be realized, in uncertainty sampling only one classifier is needed for the model (Baldridge and Osborne, 2004).

There are some ways to compute the confidence score that the classifier uses. Among them, 'entropy' proposed by Shannon (1948) is the most popular one (Zhang and Chen, 2002; Baldridge and Osborne, 2006; Chen et al., 2006; Zhu and Hovy, 2007; Zhu et al., 2008ab; Settles, 2008; 2009). Using entropy as the confidence score, the samples which have high entropy will be the most informative samples since the prediction score of the classifier for such samples is low and these samples, as a result, are the best candidates to be selected and labeled by the oracle.

3.3.2 Query-By-Committee

Seung et al. (1992) and Freund et al. (1997) proposed another query method that is widely used called 'query by committee'. In this method more than one classifier is used and each classifier tries to pick up the samples which are the most informative. Among the selected informative samples for each classifier, the samples which have the most degree of disagreements between the committee of the classifiers are selected and handed out to the oracle for labeling.

3.4 Stopping Criteria in Active Learning

In the introduction of this section, it was said that one of the most important properties of active learning is minimizing human's effort to label data; moreover, its other property s that it is an iterative process. The question is how iteratively should we continue the labeling process as the human's effort is reduced? Surely we do not

want to label all the data in active learning to terminate, so a stopping criterion should be proposed. The idea behind the stopping criteria is that the classifier stops when it has reached to its maximum effectiveness and no informative samples could be found in the data. In the followings, some of the stopping criteria are described.

It is completely obvious that it does not make sense to continue active learning process till the whole corpus is labeled by the oracle since the primer goal of active learning is reducing the labeling effort; but its positive point is having the learning curves at the end. Such learning curves provide a clue to the active learning that probable informative samples are given to the classifier to increase the learner's performance (Chen et al., 2006).

The simplest and general stopping criterion is when there is no more improvement on the performance or when the improvement is at a non-satisfactory rate. In this method the effectiveness of the classifier is evaluated (Vlachos, 2008). The other simple stopping criterion that Chen et al. (2006) has used is when the training set has reached to a desirable size.

Another stopping criterion which is statistical uses the confidence of the classifier (Zhu and Hovy, 2007). This approach which is confidence-based strategy uses the maximum confidence and the minimum error. Maximum confidence is based on uncertainty measurement when the entropy of the selected unlabeled sample is less than a predefined threshold close to zero; and minimum error is the feedback from the oracle when active learning asks for the true label of the selected unlabeled sample and the accuracy prediction of the classifier for the selected unlabeled sample is larger than a predefined accuracy threshold. These criteria are considered as upper-bound and lower-bound of stopping condition.

In another stopping criterion for active learning, since the amount of annotated data for the classifier is not clear to reach the maximum effectiveness (Lewis and Gale, 1994), a statistical learning approach called 'minimum expected error strategy' is proposed by Zhu et al. (2008a). In this approach the maximum effectiveness of the

classifier is reached when the classifier's expected errors on future unlabeled data is minimum.

3.5 Could active learning be always helpful?

In previous sections we focused on active learing and discussed about this machine learning in details and its advantages. Regarding the performance of active learing, it is believed that active leraning performs better or atleast as well as random sampling in most cases and should outperform random sampling in some circumstances. But a question might be raised whether active learning be a help for goods and outperforms the baseline.

Baldridge and Osborne (2004) believed that active learning could be used for labeling the data when the task is well-understood in terms of the computatoinal methods or the features used for the model such as POS tagging, parsing; or when small changes could be made on the method in case such methods are accepted. However, they added that it is possible random sampling be a better strategy than active learning when the method is really uncertain and not well-understood; or when there is an uncertainty on the data that has been produced in active learning process. Moreover, Dang (2004) in her Ph.D dissertation has mentioned that random sampling could be better than active learning if the the system has not seen enough context for assigning the corret label. Furthermore, she added as random sampling represents the data distribution in the corpus and active learning just picks up the samples which are difficlut for the machine to label, this causes a problem for the classifier and could result to the skewness of such informative samples on the distribution of labels in the training data which could be different from the test data.

3.6 Applications of Active Learning in NLP

Active learning has usages in various applications, but recently a special attention is given to this learning method in processing the natural language. It is worth to mention some of these applications such as: speech recognition (Hakkani-Tür et al.,

2002), spoken language understanding (Tur et al., 2003; 2005), parsing (Thompson et al., 1999; Hwa, 2000; Tang et al., 2002; Hwa et al., 2003; Steedman et al., 2003; Baldbridge and Osborne, 2003; 2004; 2006; Osborne and Baldridge, 2004; Becker and Osborne, 2005), part of speech tagging (Dagan and Engelson, 1995; Engelson and Dagan, 1996), chunking (Ngai and Yarowsky, 2000), information extraction (Thompson et al., 1999; Settles and Craven, 2008), information retrieval (Zhang and Chen, 2002; Yu, 2005), semantic role labeling (Busser and Morante, 2005; Roth and Small, 2006), document classification (Schohn and Cohn, 2000; Roy and McCallum, 2001), text classification (Lewis and Gale, 1994; McCallum and Nigam, 1998; Tong and Koller, 2001; Zhu et al., 2003; Baldbridge and Osborne, 2006; Zhu et al., 2008ab), text categorization (Lewis and Catlett, 1994; Liere and Tadepalli, 1997; Hoi et al., 2006), document segmentation (Settles and Craven, 2008), word segmentation (Sassano, 2002), word sense disambiguation (Fujii et al., 1998; Dang, 2004; Chen et al., 2006; Chan and Ng, 2007; Zhu and Hovy, 2007; Zhu et al., 2008ab), handwritten digit recognition (Zhu et al., 2003), machine translation (Haffari and Sarkar, 2009; Haffari et al., 2009), and name entity recognition (Shen et al., 2004; Laws and Schütze, 2008).

3.7 Related Works on WSD with Active Learning

From Section 2.3 we learnt that frame assignment could be considered as a WSD step in semantic role labeling; also above some NLP applications that have used active learning are mentioned including WSD. From these two, it could be concluded that somehow active learning is used for frame assignment; but not specifically as we aim to do. As there is no previous work on frame assignment with active learning, instead we present the previous studies on WSD with active learning in the followings:

Fujii et al. (1998) have used an example-based system to disambiguate verb senses in pool-based setting. In their study, they have focused on Japanese verbs to be

disambiguated with the help of a database containing example collocations for each verb sense and its associated case frame. Giving a sentence containing a polysemous verb, the system chooses the most plausible verb sense from the candidates on the basis of a similarity score between the input sentence and the examples in the database, and chooses the verb sense that is associated with the example rather than the maximized similarity score. In other words, after training their system, the samples are selected in which the system has the highest prediction score among the unlabeled data and has the highest certainty. Using the case frames as restrictions on sense selection, they have proposed a weighted case contribution. They further reported that their system performance has been more effective than other sampling methods such as random-sampling and uncertainty sampling, and query-by-committee sampling.

The drawback of their system is that the performance of their system heavily relies on the example-based learning method.

Chklovski and Mihalcea (2002) created a system called Open Mind Word Expert and they used active learning for WSD. They believed that having a high accuracy in disambiguation depends on the size of the training corpus; so that they tried to build a large corpus of sense-annotated examples from the Web users who voluntarily contributed. Their aim was creating the training data of the most frequent ambiguous words in English for SENSEVAL-3 lexical sample activity. Using stream-based active learning scenario in their task, they have utilized query-by-committee query method such that the system selected the samples having the highest degree of disagreement on the two classifiers. The first classifier was Semantic Tagger with Active Feature Selection which utilized to select features automatically, and the second classifier was Constraint-based Language Tagger which utilized WordNet to create soft constraints. In the paper no formal evaluation was presented.

Dang (2004) investigated the role of lexical semantics on WSD. The contribution to do that was defining criteria humans use to distinguish verb senses, as the target words of the study, and translating such criteria into linguistically-motivated features as computational features to build an automatic fine-grained WSD system which has used maximum entropy model. The fine-grained senses for the target English verbs were from WordNet 1.7. Firstly, to make a distinction between polysemy and homonymy, 'Levin verb classes' have been used. The idea behind Levin verb classes is that the distribution of syntactic frame in which the verb occurs determines the verb's class membership; so that it reflects the underlying semantics. In the next step, features such as syntactic frames and semantic predicates for each verb sense have been extracted from VerbNet to make the model; and finally the system has been tested on the English verbs of SENSEVAL-1 and SENSEVAL-2. It was further reported that utilizing predicate-argument features provided by ProbBank has improved the accuracy of the WSD system significantly.

To get sufficient training data for the WSD system, active learning was used to add more data with the collaboration of a real human annotator. The active learning scenario which used was pool-based with uncertainty sampling. To this aim, 5 English verbs (namely 'develop', 'dress', 'pull', 'serve', and 'treat') were chosen. BNC corpus was used to collect 500 additional data for the target verbs. The system was tested on SENSEVAL-2 test data. The final results were surprising while random sampling performed better than active learning. Even the additional samples of training data did not change the result. The proposed explanation of the findings was the quality of manually sense-tagged data which was limited by an inconsistent or unclear sense inventory for the fine-grained senses.

Chen et al. (2006) have used two uncertainty sampling methods _uncertainty sampling and margin sampling _ in pool-based active learning combined with maximum entropy model for 5 English verbs (namely 'add', 'do', 'feel', 'see', and 'work'), and compared their results with random-sampling as their baseline. They

have used course-grained senses for the target words, both to limit the impact of noisy data due to unclear sense boundaries; and to have a better understanding on the effect of active learning methods. In the study, they used 500 to 700 data set for each target verb. From the achieved results, they reported margin sampling has achieved better results. In their task, active learning was terminated when the training set has reached to a desirable size. They have done the data analysis both on the sample and feature levels; further they founded out what the good and bad samples and features are, and what their properties are. They suggested that giving a careful attention on feature extraction is important for active learning used for WSD. They have also taken overfitting phenomena into account and founded which samples occurred in the overfitting region for each verb.

Chan and Ng (2007) used active learning for domain adaptation in fine-grained WSD such that additional training data from another domain is added to the WSD system. For their study they have used DSO corpus which is composed of Brown Corpus (BC) and Wall Street Journal (WSJ) corpus. In their task, the additional training data from the BC corpus was added to the WSD system which is already trained with the WSJ corpus. The target words they focused on were nouns. As the results show, they have successfully used active learning to perform the domain adaptation for WSD.

Zhu and Hovy (2007) have studied the class imbalance problem for WSD with maximum entropy model. They have studied the effect of resampling method _over-sampling, under-sampling, and bootstrap-based over-sampling_ for WSD and compared them to uncertainty sampling and random-sampling as the baseline. In their study, they have used the WSJ part of Penn Treebank for 38 ambiguous nouns selected randomly. Based on the results, bootstrap-based over-sampling method had the best performance and random-sampling the worst. Considering other resampling methods, under-sampling had negative effects on active learning due to loosing data, but over-sampling had a relatively good results. The other discussed issue was

inventing the confidence-based stopping criteria _maximum confidence and minimum error_ for active learning.

Zhu et al. (2008a) have studied a stopping criterion for uncertainty sampling method in pool-based active learning which is called 'minimum expected error strategy' for WSD and text classification tasks with maximum entropy model. In their study, the WSJ part of Penn Treebank for 302 ambiguous nouns has been used for WSD; and WebKB corpus, which was formed by web pages gathered from various university computer science departments, has been utilized for text classification. The proposed criterion outperformed the confidence-based strategy.

Zhu et al. (2008b) have introduced two sampling methods _sampling by uncertainty and density, and sampling by clustering_ for WSD and text classification with maximum entropy model. In 'sampling by uncertainty and density' (SUD) the *K*-nearest-neighbor-based density measure was used to solve the problem of selecting outliers in uncertainty sampling. In 'sampling by clustering' (SC) *K*-means clustering algorithm was utilized to build the most representative samples in the initial training data set for active learning. In their study, Interest data set, 2369 sentences of the noun 'interest' in 6 different senses, was used for WSD task; and Camp2, consisting 'computer graphics' and 'computer Windows' categories from News Groups, and WebKB data sets were used for text classification task. They have also experimented combinations of various sampling methods and concluded that the combination of SUD and SC have the best performance compared to other methods.

3.8 Summary

In this chapter briefly machine learning methods were introduced. Representing the idea behind machine learning, active learning as a supervised learning method was described. The scenarios and query methods along with stopping criteria in active learning were discussed. In addition, some NLP applications that have used active

learning were named; and at the end, the previous studies on WSD were represented in details.

Chapter 4

Data Collection

4.1 Introduction

In previous sections, we mainly considered theoretical basis. Now that we are familiar enough with the relevant background, we focus mainly on our task which is assigning frames with the help of active learning. In this section we specifically talk about the data and data gathering.

4.2 The Data

The annotated data that we will use for frame assignment is the current version of Berkeley FrameNet (Baker et al., 1998) for English which consists of 139,437 annotated examples from British National Corpus (BNC) for 10,196 predicates. Among the predicates that FrameNet involved, namely verbs, nouns, adjectives, and prepositions, we will only consider 'verbs'; so the data reduces to 61,792 annotated examples for 2,770 unique verb-frames.

Before making the data divisions into different sets, there are some issues for the FrameNet data that should be considered: there are a number of predicates that have only one frame; for such data the accuracy of frame assignment is always 100% because no intelligence is required. Consequently, the predicates that have more than one frame are more interesting for our study and should be selected.

Considering the number of annotated samples for each frame of a verb, there are targets that have no annotated samples, so they are not useful in our study too; and they should be removed. Also, for many frames of a verb, the numbers of annotated samples are one or two. Such data again is not useful for our purpose since the data will be divided into three sets, as we will see below in Section 4.4, at least three samples for each frames of the predicate are required.

Considering these issues, the numbers of predicates are reduced to 451 unique verbs having more than one frame and having more than three annotated samples. The other parameter that should be considered is the frequency of the predicate. In our study we plan to choose the verbs that are quite frequent having annotated samples above 100 as a whole for different frames. The target verbs are given in Table 3:

Table 3. List of 37 verbs with their relevant numbers of frames and samples

VERB	Frames	Total Samples	VERB	Frames	Total Samples
Admire	2	116	predict	2	109
Argue	3	104	push	2	110
Bend	4	101	reach	4	100
Break	4	115	rise	4	110
Crawl	2	147	rush	2	168
Discuss	2	107	scream	2	148
Drop	4	136	see	2	127
Escape	3	108	shake	4	104
Express	2	136	shuffle	2	105
Fall	2	127	smack	3	101
Feel	5	134	smell	3	146
Follow	3	113	steal	2	110
Forget	3	101	strike	3	105
Hear	2	124	tell	2	151
Hit	4	142	throw	2	155
Increase	2	122	understand	3	135
Lean	3	103	urge	2	157
Look	3	183	withdraw	3	125
Phone	2	166			

Among these verbs, we mainly do our experiments on 14 verbs which were selected randomly; however, in the selection we tried to have a balance distribution of targets in terms of the number of frames they have. As a result, 4 targets (‘phone’, ‘rush’, ‘scream’, ‘throw’) having two frames, 5 targets (‘follow’, ‘forget’, ‘look’, ‘smell’,

‘strike’) having three frames, 4 targets (‘bend’, ‘hit’, ‘rise’, ‘shake’) having four frames, and 1 target (‘feel’) having five frames were selected.

4.3 Data Preparation

The FrameNet data is in XML format; so that the 5 types of information described in Section 2.2 are available for each annotated sample. The relevant samples of a frame with respect to its target predicate are in one XML file. For our task, we have to remove the irrelevant information and only extract the most useful information which is the samples along with their frame names. Using Perl scripts, such data is converted into the plain text format; as a result, we have several text files containing all the samples and the labels. Then, we try to merge the samples of the same target predicate into a single text file. The result is 14 text files for our 14 target verbs.

4.4 Data Distribution

Since active learning is a kind of supervised learning method, we require a set of annotated date for training which should be big enough. We also need a set of annotated data as test data to test the model and compare the results with respect to the chosen active learning method; so that we could accept either of the proposed hypotheses in Chapter 1.

The total amount of data we prepared for 14 verbs should be divided into three non-overlapping sets in a balanced form in terms of both the number of frames of the target predicate, and the relevant samples of each frame. In other words, the distribution should be in a way that different frames of the target verb be found in each of the three data sets. 10% is considered as initial seed data; 20% as test data, and the rest of 70% as extra unlabeled data. The data distribution is given in Table 4.

Looking carefully at the data distribution, we see that there is at least one frame for a verb which is skewed over other frames; i.e. different frames of verbs, often, do not have equal numbers of instances. Such problem is called ‘imbalance problem’. We will have more details about this problem in Section 5.3.2.

Table 4. Data distribution for target verbs with respect to their frames

Verb	Frames	Total Samples	Seed Data	Extra Unlabeled Data	Test Data
Bend	Body_movement	51	5	36	10
	Change_posture	32	3	23	6
	Path_shape	4	1	2	1
	Posture	14	2	9	3
Feel	Appearance	6	1	3	2
	Feeling	49	4	36	9
	Perception_active	14	2	9	3
	Perception_experience	60	5	44	11
	Seeking	5	1	3	1
Follow	Compliance	20	2	14	4
	Cotheme	75	6	55	14
	Relative_time	18	2	12	4
Forget	Remembering_experience	34	3	24	7
	Remembering_information	33	3	24	6
	Remembering_to_do	34	3	24	7
Hit	Cause_harm	74	6	54	14
	Cause_impact	12	1	8	3
	Experience_bodily_harm	11	1	8	2
	Impact	45	4	32	9
Look	Appearance	49	4	36	9
	Perception_active	109	9	80	20
	Scrutiny	25	2	18	5
Phone	Communication_means	6	1	3	2
	Contacting	160	13	118	29
Rise	Change_posture	37	3	27	7
	Motion_directional	9	1	6	2
	Path_shape	13	2	8	3
Rush	Fluidic_motion	19	2	13	4
	Self_motion	149	12	110	27
Scream	Communication_noise	73	6	53	14
	Make_noise	75	6	55	14
Shake	Body_movement	48	4	35	9
	Cause_to_move_in_place	18	2	12	4
	Experiencer_obj	23	2	16	5
	Moving_in_place	15	2	10	3
Smell	Appearance	87	7	64	16
	Perception_active	15	2	10	3
	Perception_experience	44	4	32	8
Strike	Cause_harm	49	4	36	9
	Cause_impact	15	2	10	3
	Impact	41	4	29	8
Throw	Body_movement	34	3	24	7
	Cause_motion	121	10	89	22

Since our data set is too small and we have the sparse data problem, it is hard to have a strong conclusion out of the results; so that to have a better result representation and make it possible to draw conclusions, and also to minimize the overfitting phenomena, we have used 5 fold cross-validation.

In such data distribution, we have split the whole data into 5 equal parts so that each part contains 20% of the whole data. Then, each split part is considered as the test set for one fold; as a result the test set of the 5 folds is not overlapping and equal. For the rest of 80% data, the first 10% is regarded as initial seed data and the remained 70% as extra unlabeled data. In our study, active learning is run on each fold and the average performance of each iteration for 5 folds would be considered as the final result of the relevant iteration.

4.5 Summary

In this section we mainly focused on the data that we will use for our experiments. Among the predicates in FrameNet, we only considered the data for 14 frequent verbs. Furthermore, the data distribution was discussed such that the data of each target verb was divided into three sets. Since the data set was small, 5 fold cross-validation was used to have a more reliable results and minimize overfitting phenomena.

Chapter 5

Methodology

5.1 Introduction

In the previous section, we paid a special attention on the data we want to use in our experiments. This section devotes to the algorithm we use for frame assignment with the help of active learning in details. Moreover, we will have a closer look at the proposed model and its components.

5.2 Proposed Model

Considering the description we had on shallow semantic parsing and frame assignment in Section 2.2, and active learning in Section 3.1, we aim to use the Supervised Learning Semantic Processor (SLSP) toolkit described below for our purpose which is frame assignment with the help of active learning. To our knowledge, frame assignment with the help of active learning is not previously studied specifically; and the FrameNet data set, particularly, has not been used in the previous research on WSD with active learning. Based on all these, we can build the architecture of our model as represented in Figure 3.

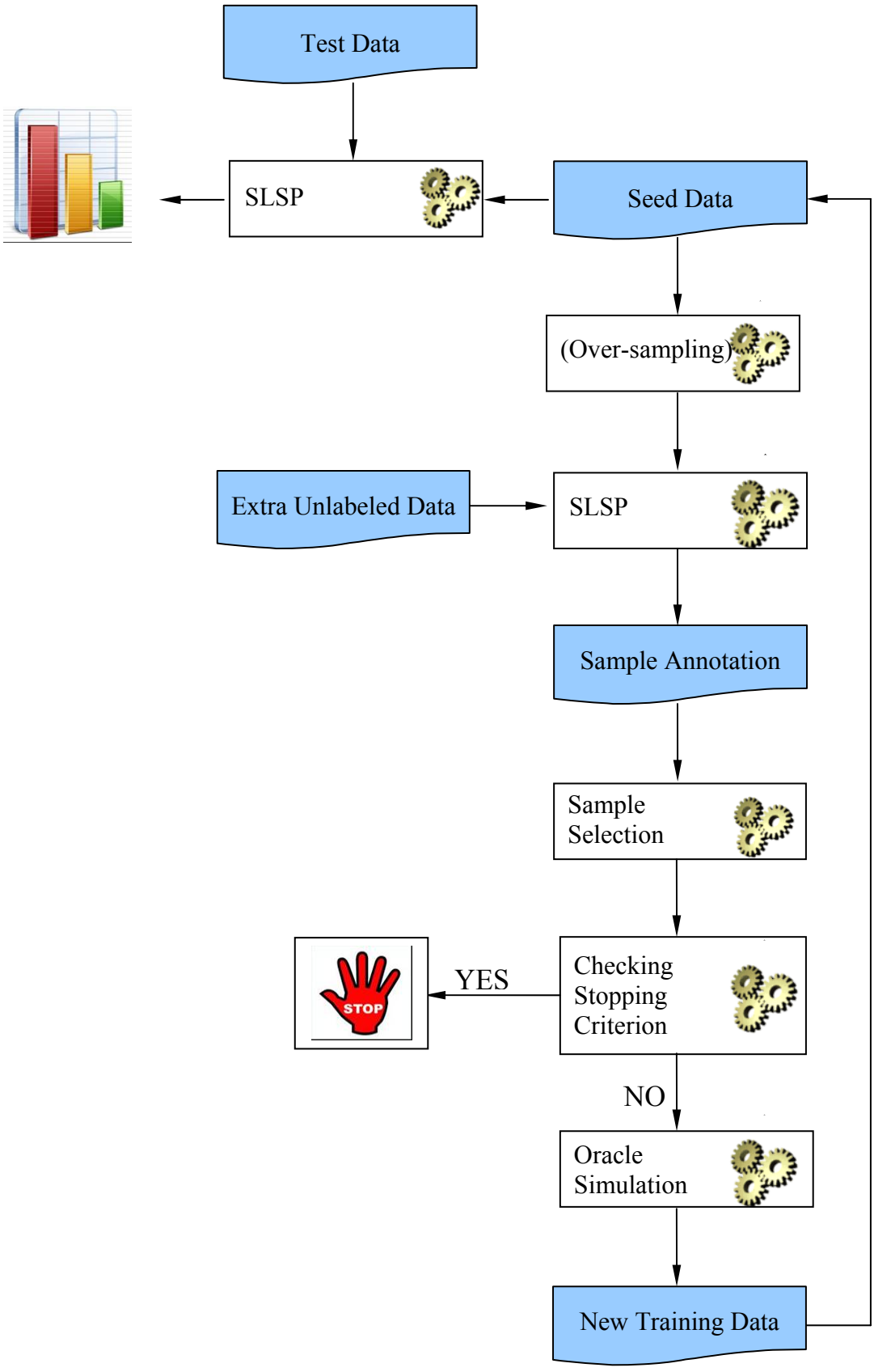

Figure 3. The architecture of our model

5.3 Components of the Model

Considering the architecture of our proposed model in Figure 3, it is composed of 5 major components in which we will have a closer look at each of them below.

5.3.1 SLSP Tool

SLSP is a toolkit which has a graphical user interface (GUI) for active learning to do semantic annotation; however, it could be used for predicting labels of unlabeled data, further training the system, and evaluating the performance of the system. The toolkit supports German and English; and it uses the openNLP MAXENT package[*], a Java implementation of maximum entropy classifier, to build the model. The toolkit implements the uncertainty sampling query method for stream-based active learning. As we know from Section 3.3.1, we require a confidence score for uncertainty sampling to measure the informativeness of the training data. In this toolkit, the confidence score of the classifier is the posterior probability of the most probable label assigned to each sample.

Among the properties of the tool, one is setting a threshold manually to select the uncertain samples. Having defined the threshold *K*% in each iteration, the toolkit will return the samples which have the confidence below the threshold to the oracle. The other property is retraining the system with additional annotated samples, called 'further training'. In this step, the new labeled instances will be added to the system and the system will be retrained by the instances already added and the newly labeled instances.

In the tool, there are some built-in plugins for syntactic and semantic pre-processing to provide the relevant features for the classifier. The order of the extracted features is crucial to train the classifier. In the followings, the plugins that support English and used in our study would be described briefly along with the type of information they extract for the classifier. To have a sense what type of information is extracted,

[*] http://maxent.sourceforge.net

it worth to show an instance with the information that the plugins provide. As shown in the sample below, the first line is the instance, the second line the frame of the instance, and the followings lines firstly the name of the plugin is written then the relevant information that the plugin provides. It should be pointed out that in SLSP features are indexed as C*i* so that it makes the position of the words surrounding the target word identical.

```
TARGET ' I 'd almost forgotten how good you look in the mornings . "
FRAME  Remembering_experience
       WordRangeClusterer [1.0 [StanfordPosTagger]]
             almost 'd I how good you
       StanfordPosTagWordRangeClusterer [1.0 [StanfordPosTagger]]
             RB NNP NNP WRB JJ PRP
       SentencePhraseClusterer [1.1 [BerkeleyParser]]
             'd almost forgotten how good you look in the mornings
       SentencePhrasePosTagClusterer [1.1 [BerkeleyParser]]
             VBD RB VBN WRB JJ PRP VBP IN DT NNS
```

Plugin 1: Stanford Word Range Plugin
(`WordRangeClusterer`**)**

The Stanford Word Range Plugin, which uses Stanford POS Tagger, tries to provide features based on the local context of the surface string. The window size of the local context can be set manually in the GUI. To find out what the best window size is, we have done some initial experiments for the target verbs and we found out window ±3, 3 words before and 3 words after the target, has a better performance than a smaller or bigger window size.

For the above instance, C0=`I` (position -3), C1=`'d` (position -2), C2=`almost` (position -1), C3=`how` (position +1), C4=`good` (position +2), and C5=`you` (position +3) will be extracted and used as features for the classifier.

Plugin 2: Stanford POS Tag Word Range Plugin
(`StanfordPosTagWordRangeClusterer`**)**

The Stanford POS Tag Word Range Plugin, which uses Stanford POS Tagger, provides the POS tags of the words within a sentence. In this plugin, it is also possible to set a window size to extract the POS local context of the target word. To

find out what the best window size is, again, we did some initial experiments and we found out window ±3, 3 words before and 3 words after the target, has a better performance than a smaller or bigger window size.

For the above instance, C6=NNP (position -3), C7=NNP (position -2), C8=RB (position -1), C9=WRB (position +1), C10=JJ (position +2), and C11=PRP (position +3) will be extracted and used as features for the classifier. It should be pointed out that in our study, we did not want this plugin to tag punctuation marks.

Plugin 3: Berkley Sentence Phrase Plugin
(SentencePhraseClusterer**)**

The Berkley Sentence Phrase Plugin, which uses Berkley Parser, provides the syntactic analysis of the sentence. This plugin is used to identify and extract all word forms of the children nodes from a particular syntactic mother node (VP in our study) and add them to the feature set.

For the above instance, C12='d, C12=almost, C12=forgotten, C12=how, C12=good, C12=you, C12=look, C12=in, C12=the, and C12=mornings will be extracted and used as features for the classifier. As we can see, the index of all features for this plugin is the same, which means the position and the order of the words provided by this plugin are not important for the classifier.

Plugin 4: Berkley Sentence Phrase POS Tag Plugin
(SentencePhrasePosTagClusterer**)**

The Berkley Sentence Phrase POS Tag Plugin uses Berkley POS tagger such that we define the mother node of the target word in the parse tree (VP in our study) and it identifies and extracts all children of this mother node and uses their POS as features.

For the above instance, C13=VBD, C13=RB, C13=VBN, C13=WRB, C13=JJ, C13=PRP, C13=VBP, C13=IN, C13=DT, and C13=NNS will be extracted and used as features for the classifier. Again, the same as plugin 3, the position and the order of the words provided by this plugin are not important for the classifier.

5.3.2 Active Learning

Among the different scenarios for active learning available in the literature which have been utilized for NLP applications, researchers such as Cohn et al. (1994), Dagan and Engelson (1995), Fujii et al. (1998), and Yu (2005) have used stream-based scenario; and Lewis and Gale (1994), McCallum and Nigan (1998), Thompson et al. (1999), Tong and Koller (2001), Tur et al. (2005), Hoi et al. (2006), Chen et al. (2006), and Settles and Craven (2008) have used pool-based scenario. Considering our task, we decided to choose the 'pool-based' scenario as well to have nice and uniformed learning curves for the output, and also make it possible to compare the curves. Since the SLSP tool does not rank the uncertain samples to select the top *K* samples in the pool-based mode, we have provided Perl scripts for this purpose.

Uncertainty sampling

For the query strategy of active learning, following Baldridge and Osborne (2004) who have indicated most studies on active learning in NLP have primarily focused upon uncertainty sampling which produces good results in comparison to other active learning instance selection methods, and other people like Lewis and Gale (1994), Thompson et al (1999), Hwa (2000), Tang et al (2002), Chen et al (2006), and Zhu et al (2008ab) who have used the uncertainty sampling as their first query strategy, consequently, we have taken the advantage of using this query method in SLSP toolkit. The algorithm of active learning (Chen et al., 2006) with uncertainty sampling could be found in Figure 4:

Procedure: Active Learning with Uncertainty sampling

Input: Initial seed data *S*, Pool of unlabeled samples *U*
    Use *S* to train the classifier *C*
While the stopping criterion is met **do**
    Use *C* to annotate *U*
    Select top *K* samples from *U* predicted by *C* which have the lowest confidence
    Label *K*, augment *S* with *K* samples, and remove *K* from *U*
    Use *S* to retrain *C*
End while

Figure 4. Active learning algorithm with uncertainty sampling

Considering the algorithm, in each iteration 5 samples ($K = 5$) which have the lowest confidence score will be selected, labeled by the oracle, added to the training data, and removed from the unlabeled pool of samples.

Resampling

Recalling from Section 4.4, we have the imbalance problem in our data set. Having this problem, the classifier gives different prior probabilities to the frames, so the learner over-predicts the majority class in comparison to the other classes. The solution that we have thought of is using 'resampling method' in our model; so that all classes will have the same frequency in the training data. There are two resampling methods, namely 'over-sampling' and 'under-sampling' (Japkowicz, 2000; Zhou and Liu, 2006; Zhu and Hovy, 2007). In over-sampling the class which is underrepresented will contain as many examples as the class which is overrepresented. One of the methods for over-sampling is selecting samples randomly from the minority class and copying them to the same class until reaching to the same level of the majority class. In under-sampling some samples of the majority class will be removed randomly to reach the same level of the minority class.

In our model, we have used uncertainty sampling with over-sampling method for frame assignment to minimize the impact of biasing the classifier having the imbalance problem. The advantage of over-sampling is that the classifier will have an equal prior probability to the labels belonging to either the minority or the majority class. Then, the results of over-sampling method in active learning will be compared to the active learning with ordinary uncertainty sample selection. That is the reason in Figure 3 over-sampling is in parenthesis to indicate it is not used in all models.

It is expected to have a more effective classifier in over-sampling than the classifier trained on ordinary uncertainty sampling since the effect of skewness of one frame over the other one is reduced by considering prior probability of the classes

uniformed; so the prior probability will not have any influence on the classifier's label assignment.

The algorithm of active learning with over-sampling (Zhu and Hovy, 2007) is represented in Figure 5. Considering the algorithm, for our study, 5 samples ($K = 5$) which have the lowest confidence score will be selected in each iteration, labeled by the oracle, over-sampled, added to the training data, and removed from the unlabeled pool of samples.

In our task, we have done over-sampling by exact copying of samples selected randomly from the minority class(es). Since it is possible to have no sample from the minority class(es) in the selected top *K* samples, we benefit from the samples that have already been used for training.

It should be added that in our task, we have done an artificial over-sampling such that over-sampling is not done entirely randomly to have a uniform distribution among the samples selected randomly. To this end, each selected sample is checked and allowed to copy if it has not already been copied. Reselection of a sample is only allowed when all samples have been selected before.

Procedure: Active Learning with Over-sampling
Input: Initial over-sampled seed data *S*, Pool of unlabeled samples *U*
Use over-sampled *S* to train the classifier *C*
While the stopping criterion is met **do**
Use *C* to annotate *U*
Select top *K* samples from *U* predicted by *C* which have the lowest confidence
Label *K*, over-sample *K*, augment *S* with over-sampled *K*, and remove *K* from *U*
Use *S* to retrain *C*
End while

Figure 5. Active learning algorithm with over-sampling

Random Sampling

In order to find out the answer to our hypothesis how effective active learning is and to see how well the performance of our model is, our proposed model with uncertainty sampling should be compared with a baseline which is random sampling as it represents the distribution of frames in corpus. In uncertainty sampling,

instances which are the most informative would be selected and handed out to the oracle to be labeled; while in random sampling, there is no such intelligence and the instances are selected by chance and handed out to the oracle to be labeled.

5.3.3 Stopping Criteria

The stopping criteria for active learning have already been discussed in Section 3.4. Considering the aim behind active learning, the best stopping point is when the classifier has reached to its maximum performance. Achieving this point is very difficult experimentally; however, we should try to find a criterion to stop active learning in a near-optimum point.

To this aim, we have proposed a new stopping criterion which uses variance on the classifier's confidence score to represents the degree of spreading out the confidence scores around their mean.

Generally, a stoping criterian could be either based on the performance or confidence score of the classifier on test data, or based on the confidence score of the classifier on unlabeled data. In our method, we have used the second option. The most advantages of this approach is that we do not need to test the system in each iteration and also there is no overfitting on test data.

According to the pool-based senario, in each iteration K samples of the extra unlabeled data which have the lowest confidence score will be selected and after labeling by the oracle they are added to training data. In the early iterations, the mean of the classifier's confidence scores for the selected samples is low. Since the classifier is not trained enough in these iterations, most of the scores are low and they do not have a high degree of variability. As a result the variance of the confidence score for these samples would be low.

As the classifier is training with more data, the confidence score of samples would gradually increase; as a result, there would be a high degree of variability in the confidence scores which spread out around their mean. In these iterations, the classifier is somehow in the borderline of training, passing from untrained to trained;

so there would be a variability of confidence scores which leads to have a high variance.

When the classifier is trained, the confidence score of the classifier on selected samples would be increased. However, from a certain point that the classifier is trained enough, all of the confidence scores are located tightly around their mean with a low degree of variability; as a result, the variance of the samples decreases. We believe the best stopping point is when variance passes its peak and starts to decrease. Figure 6 represents the behavior of the variance on different iterations. In which the *x* axis is the number of iterations and the *y* axis is the variance of the confidence scores in each iteration.

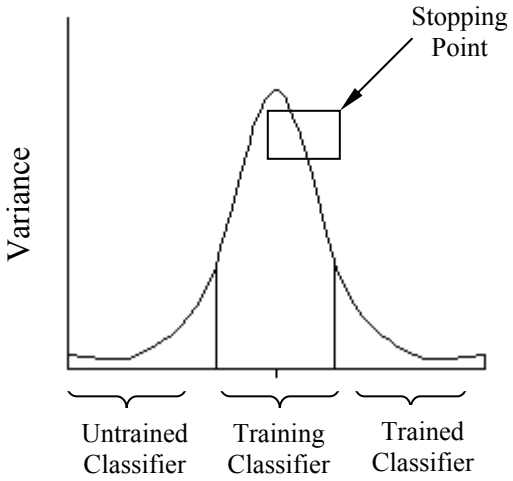

Figure 6. Normal distribution of variance for the classifier's confidence score

To compute variance in each iteration we have used this formula:

$$Variance = \frac{\sum_{i=1}^{K}(C_i - M)^2}{K}$$

where

C_i is the confidence score of each selected sample in each iteration, M is the mean of confidence score for these samples, and K is the number of samples selected in the same iteration (K =5 in our study).

We will discuss more about this stopping criterion in Section 6.3.

5.3.4 Simulated Oracle

In active learning there is an interaction between the system and the human annotator to label the informative samples. In our study, we have limited ourselves to the FrameNet data set for both training and testing. We know that using only the labeled data of FrameNet makes restrictions for us such as accessing to a limited number of labeled samples; however, using new unlabeled data and labeling them by a real human annotator is too costly. Because of having access to the gold labels in the training data, we have simulated the human annotator instead; as a result the system itself finds and assigns the correct labels of the instances based on the gold labels that it has. It should be added to have the pool of extra unlabeled data, we have ignored the labels of this data set from the original data and assumed them unlabeled.

In the three sampling methods in our model, when certain numbers of samples are picked up (5 in our study) from the extra unlabeled data, they will be labeled automatically with respect to the labels they have in FrameNet. In other words, for each selected sample, the correct frame will be searched in the original data in FrameNet. Having found the label, it will be assigned to the selected unlabeled instance to further train the system.

5.3.5 Evaluation

After introducing our tool for active learning, we need to evaluate the performance of the learing methods. To this aim, we have used the major evaluation metrics such as accuracy, precision, recall, and F-score with respect to the Table 5 considering the output of the system and the gold true labels:

Table 5. Table of confusion

		System	
		P	N
True Labels	P	tp	fn
	N	fp	tn

Where in our case

P is the target frame and *N* is other frame(s); *tp* is the number of instances which are assigned the target frame correctly; *fp* is the number of instances which are assigned the target frame incorrectly; *tn* is the number of instances which are assigned other frames correctly; *fn* is the number of instances which actually belong to the target frame but classified incorrectly to other frames.

Accuracy: This is a basic evaluation measure as the ratio of the number of correctly assigned frame (*tp* and *tn*) to the total number of the input instances having various frames (*tp*, *fp*, *tn*, and *fn*):

$$Accuracy = \frac{tp + tn}{tp + fp + tn + fn}$$

Precision: This is the ratio of the number of instances which are assigned the target frame correctly (*tp*) to the number of all instances which are assigned this target frame either correctly (*tp*) or incorrectly (*fp*):

$$Precision = \frac{tp}{tp + fp}$$

Recall: This is the ratio of the number of instances which are assigned the target frame correctly (*tp*) to the number of all instances which actually belong to the target frame; no matter they are classified correctly to the target frame (*tp*) or classified incorrectly to other frames (*fn*):

$$Recall = \frac{tp}{tp + fn}$$

F-score: This is a combined recall-precision score as a summary statistics of the precision and recall:

$$F - score = \frac{\beta \times Presision \times Recall}{Precision + Recall} \quad (\beta=2)$$

5.4 Summary

In this section, we mainly concentrated on the approach we have taken in our study, and we elaborated the algorithms used in our model. Since SLSP tool was used in our model, the features that have been extracted with the plugins and used by the classifier were briefly discussed. We further talked about the types of active learning scenario and the query strategies used in our model including uncertainty sampling, over-sampling to solve the imbalance problem, and random-sampling as the baseline. Moreover, a stopping criterion for active learning for our study and the evaluation metrics were also presented.

Chapter 6

Data Analysis

6.1 Introduction

In the previous chapter we mainly focused on our proposed model and we talked about its components. Doing experiments with the model on the data we had from Chapter 4, we will have the results in this chapter and discuss about them.

6.2 Active Learning Results

As represented in the architecture of our model in Figure 3 (page 36), to have the learning curves which represents the performance of the system with respect to sampling methods of each target, we test the system on test data after training the system in each iteration. Having such learning curves makes it possible to compare the different sampling methods at one glance. Since in our study we have used the 5 fold cross validation, the experiments for each of the sampling methods are repeated 5 times such that in each iteration of each fold, 5 instances are selected and added to the training data. Then we have represented the average performance of 5 fold for each iteration in the learning curves. The result is having 3 learning curves for the 3 sampling methods (random sampling, uncertainty sampling, over-sampling). To know how good the classifier has performed in each sampling methods, we required

a baseline to represent the minimum performance when no learning method is used. We call this baseline 'majority class baseline'. To have this baseline, we have assigned the dominant frame which belongs to the majority class to all test data and then evaluated the performance. The result gives us a line which represents the minimum performance when no learning technique is used. If the results of the sampling methods are above this baseline, it shows that the classifier has a good performance on the sampling methods and there is a positive impact on the classifier; but if it is below the baseline, it indicates the machine learning algorithm can not be a help. The learning curves of 7 targets are shown below:

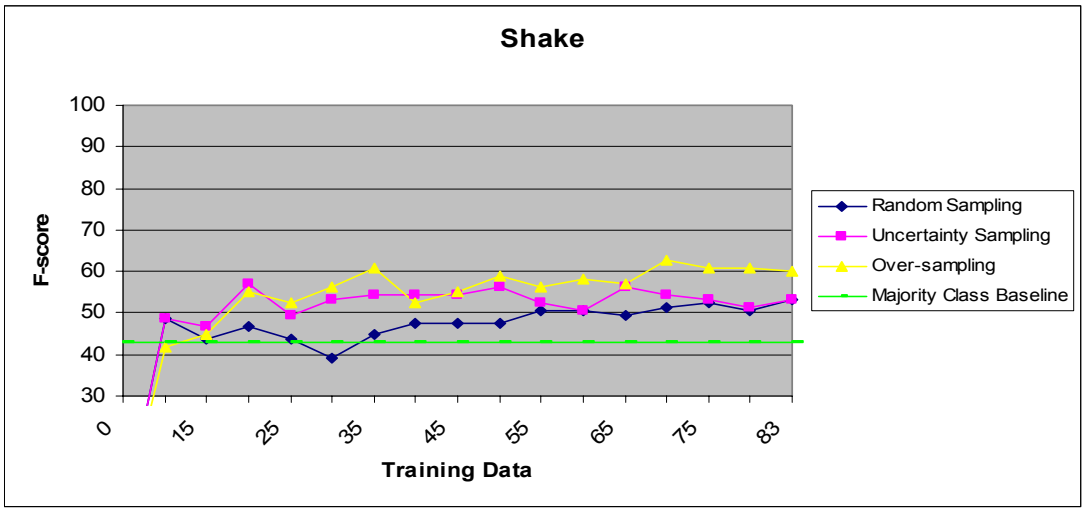

Figure 7. Learning curve of the verb 'shake'

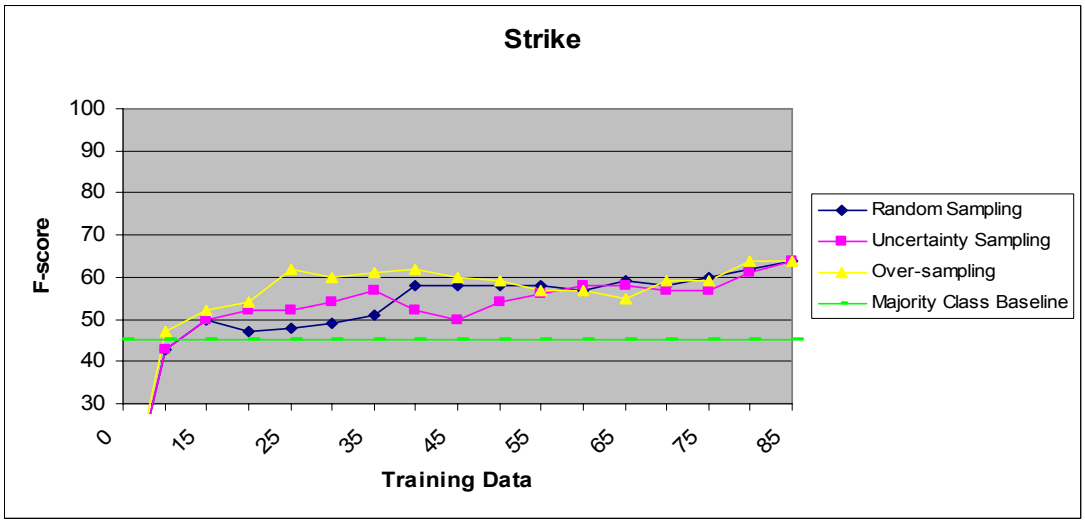

Figure 8. Learning curve of the verb 'strike'

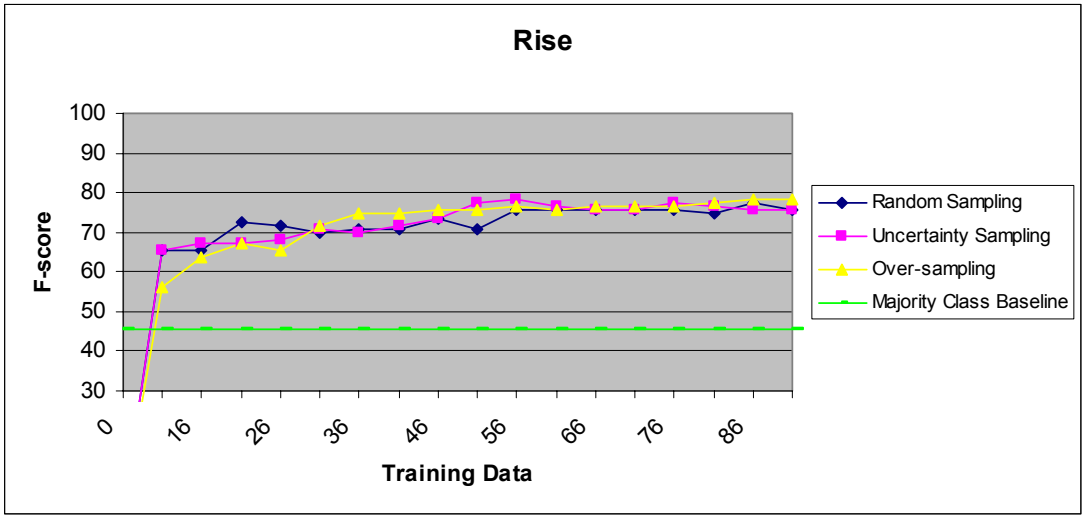

Figure 9. Learning curve of the verb 'rise'

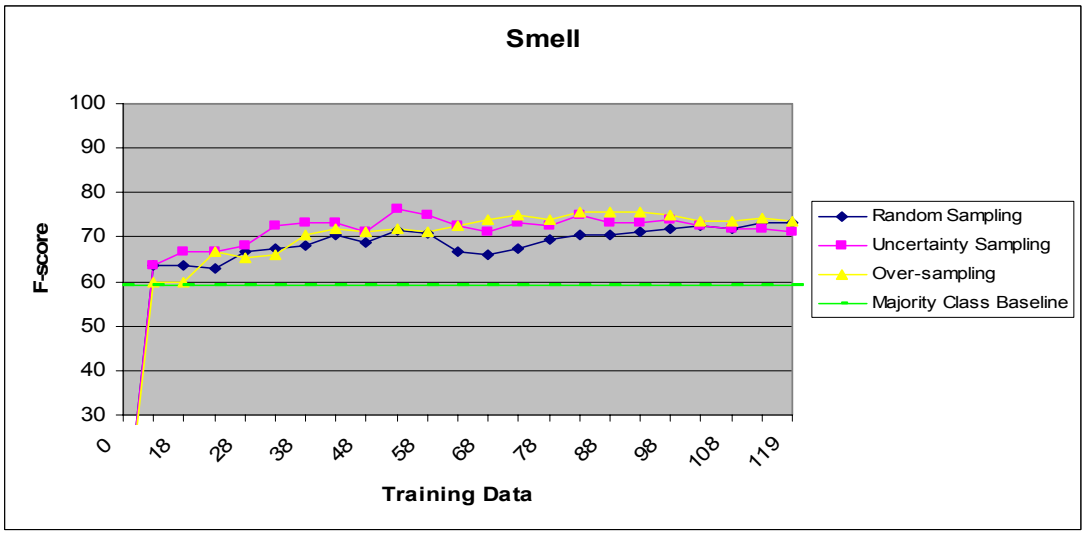

Figure 10. Learning curve of the verb 'smell'

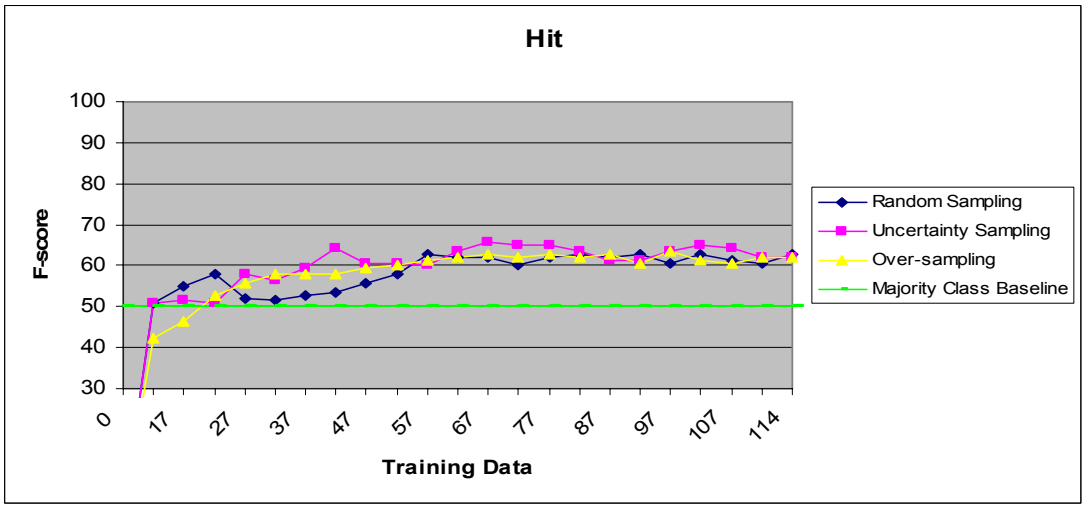

Figure 11. Learning curve of the verb 'hit'

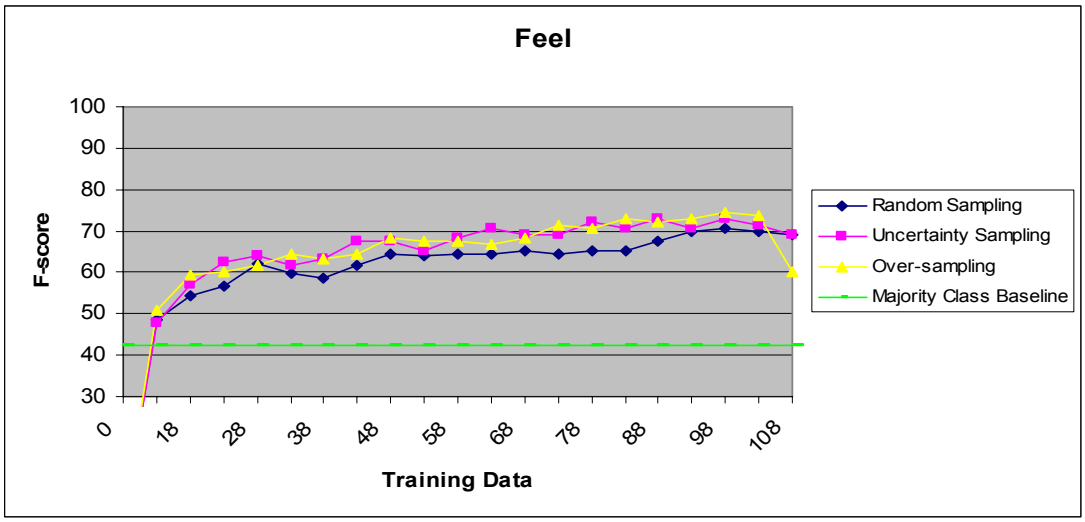

Figure 12. Learning curve of the verb 'feel'

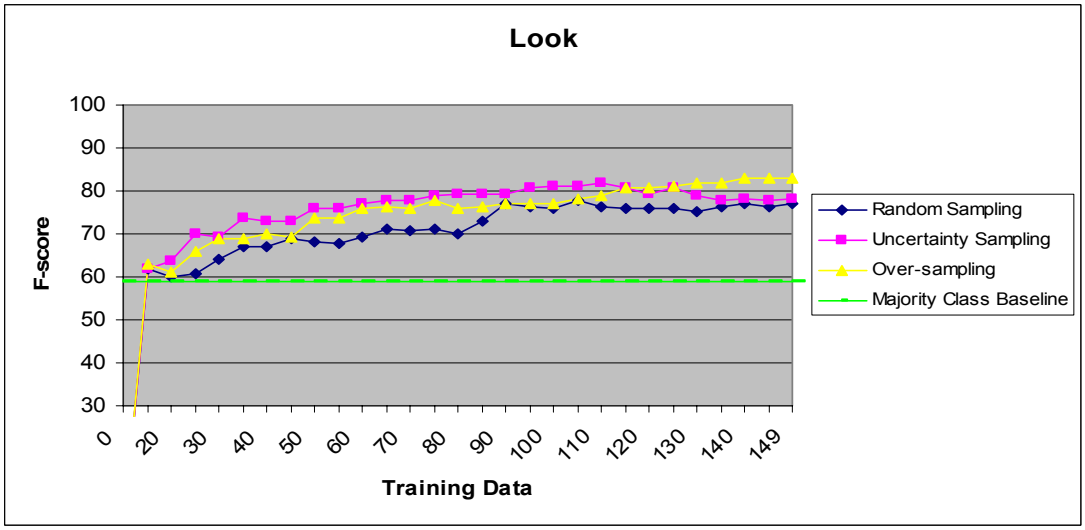

Figure 13. Learning curve of the verb 'look'

As shown in Figures 7-13, all in all, the three sampling methods have a better performance than the majority baseline; so, it determines the usefulness of machine learning approaches. As shown in the curves, we have reached a better performance with uncertainty sampling in the targets 'shake', 'strike', 'rise', 'smell', 'hit', 'feel', and 'look' compared to their relevant random sampling which indicates minimizing human's effort for labeling the data even though in our experiments the data set used for training and testing was very small and we had the sparsity problem. Having the results determines that our proposed hypothesis in using active learning for frame assignment could be true.

As represented in Figures 7 and 8, over-sampling has outperformed uncertainty sampling; and in Figure 9 even though random sampling had a better performance in the very initial iterations, it dropped; and as over-sampling performed better it crossed both random and uncertainty sampling and had a better performance than them. As already mentioned, due to having skewness on the dominant frame which belongs to the majority class in the training data for uncertainty sampling, we tried to resolve the bias of the classifier with over-sampling; so that the prior probability of all labels would be equal and the dominant frame can not influence the classifier's decision.

But it is not true in all cases as shown in Figure 10-13 in which over-sampling did not have a better performance compared to their relevant uncertainty sampling. We believe when skewness between different frames of the distributed data is high, over-sampling could not be much help. The reason is so obvious: when over-sampling is done on frames belonging to the minority class, the classifier is only limited to a set of extracted features which are used to label similar data; when the features of test data differs from features of training data, the classifier might malperform. The other possibility of malperformance of the classifier is that since in over-sampling we do the exact copying, we overfit such data in the training data. As a result, although by over-sampling the data we can get a balanced prior probability of different frames, the model may overfit to the training data, so it malperforms while evaluating on test data.

Considering the verb 'look', it has three frames such that over-sampling should be done on two of the frames belonging to the minority class. Since the majority class belongs to the frame 'PERCEPTION_ACTIVE' (consisting 89 samples as unlabeled data and 20 samples as test data), and frames 'APPEARANCE' and 'SCRUTINY' belong to the minority class (consisting 40 and 20 samples as unlabeled data, and 9 and 5 samples as test data respectively), over-sampling such samples results in having the exact copying for 49 samples of the frame 'APPEARANCE' and 69 samples of the frame 'SCRUTINY' which are added to the training data without a

change on the features that have already been extracted and the classifier trained with. Exact copying of such samples in training data results in overfitting the model on training data.

While this is not true for the verb ‘shake’ as represented in Figure 7 since the majority class has ‘BODY_MOVEMENT’ frame (with 39 samples as unlabeled data and 9 samples as test data) and the minority class has ‘EXPERIENCER_OBJ’, ‘CAUSE_TO_MOVE_IN_PLACE’, and ‘MOVING_IN_PLACE’ frames (consisting 18, 14, and 12 samples as unlabeled data and 5, 4, and 3 samples as test data respectively) such that by applying over-sampling on these three frames, 21, 25, and 27 samples of the frames should be copied which is not too high to face the overfitting problem. Comparing the proportion of over-sampled data for the verbs ‘look’ and ‘shake’, we can concluded there is a better results on the performance of the classifier when over-sampling is not high.

From the results of other target, namely ‘throw’, ‘rush’, ‘phone’, ‘follow’, and ‘bend’, we found out that neither of sampling methods could beat the majority class baseline; so there no significant improvement by using a classifier for such targets. We could not find a specific reason for these verbs; but we think when the distribution of instance for each frame of the targets is highly skewed it is really hard to beat the majority class baseline as the effect of active learning could be realized for targets ‘phone’, ‘rush’, and ‘throw’ which have two frames. The learning curves of these three targets are shown in Figures 14-16. From Table 4, we have the following data distribution for different frames of these targets:

Verb	Frames	Total Samples
Phone	Communication_means	6
	Contacting	160
Rush	Fluidic_motion	19
	Self_motion	149
Throw	Body_movement	34
	Cause_motion	121

As observed, one frame has a high degree of skewness over the other one; as a result, it is very difficult for the classifier to outperform the majority class baseline.

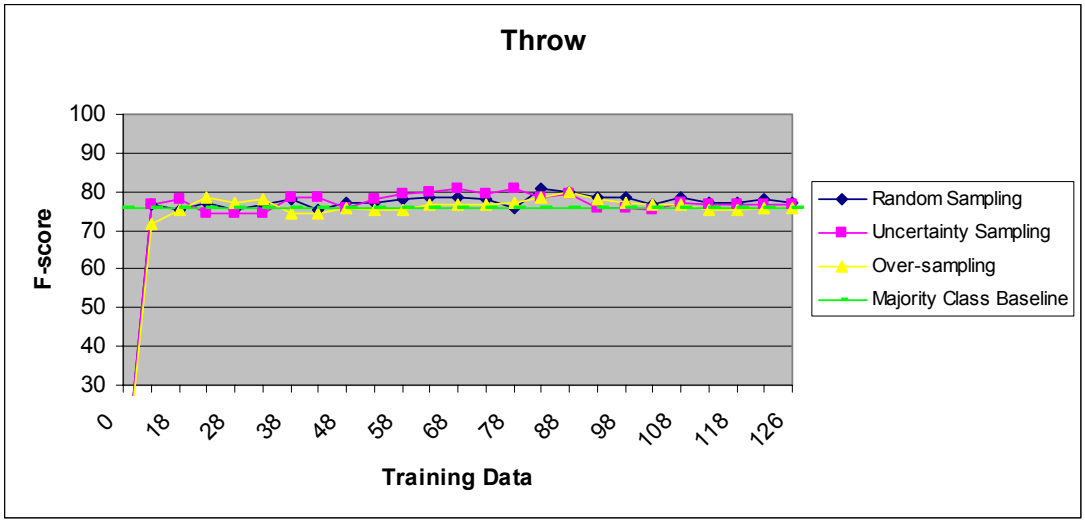

Figure 14. Learning curve of the verb 'throw'

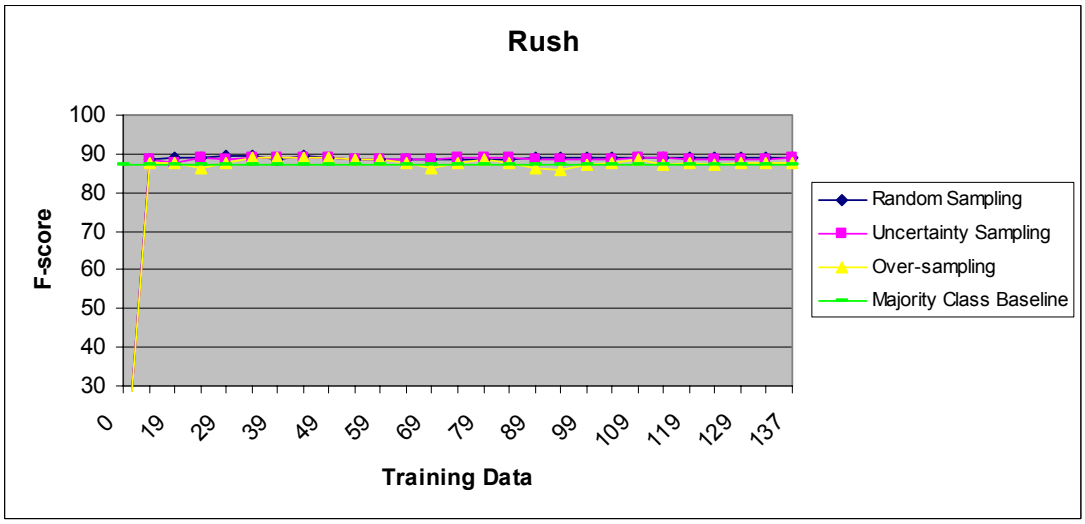

Figure 15. Learning curve of the verb 'rush'

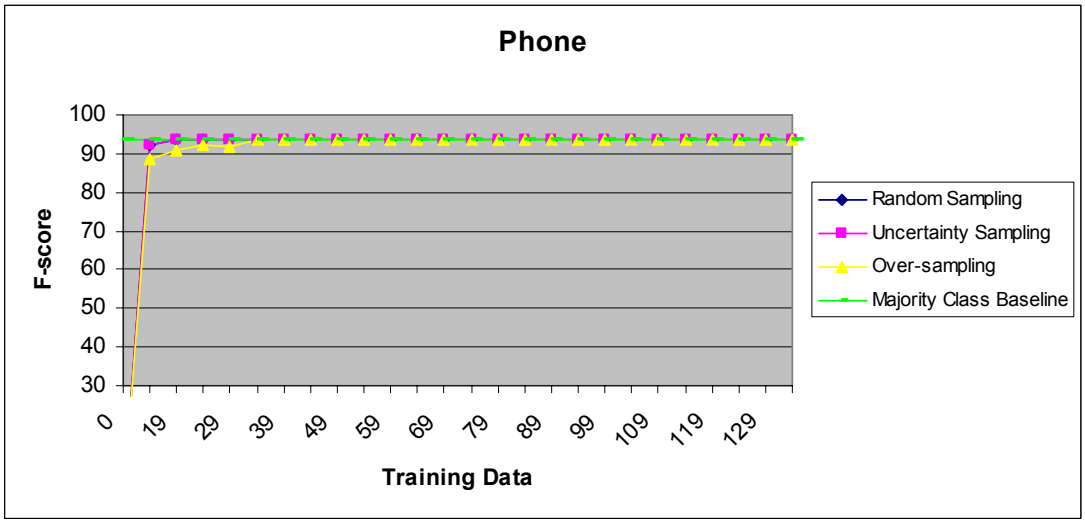

Figure 16. Learning curve of the verb 'phone'

For the other two targets, ‘follow’ and ‘bend’, active learning does not have a good performance too. As represented in Figures 17, the classifier performs better than the majority class baseline, but there is no significant difference between the three sampling methods. For the target ‘bend’, Figure 18, the result of the active learning in the initial iterations is worse than the majority class baseline. However, as the classifier is trained with more data, the performance improves as well; especially in the middle iterations where active learning had a slight better performance than random sampling. In this target, random sampling performs quite contrary such that it performs the best in the initial iterations, but the performance decreases when the system is trained with more data.

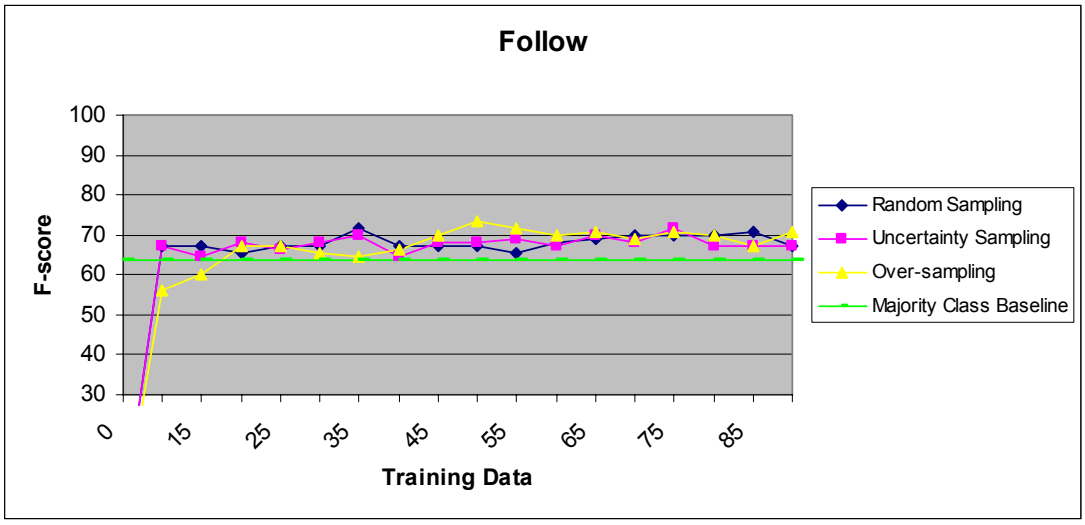

Figure 17. Learning curve of the verb ‘follow’

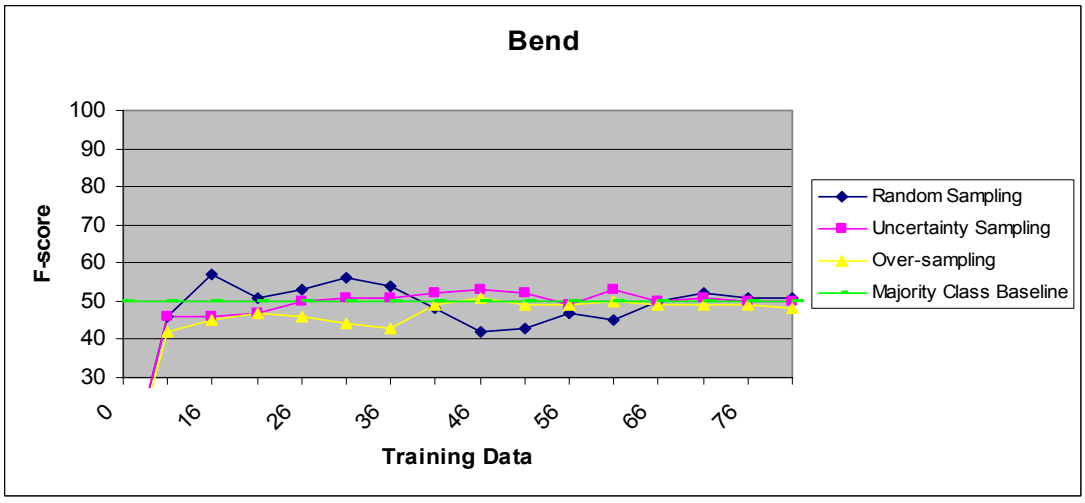

Figure 18. Learning curve of the verb ‘bend’

6.3 Stopping Criteria

Recalling from Section 5.3.3 where we introduced the new stopping criterion, we present details in this section to know how getting to the near-optimum point to stop active learning. As described, we determined how variance on classifier's confidence score is used as a key to stop active learning. We also proposed that the best stopping point is when variance passes its maximum, the global maxima. To find out how effective the stopping point is, we divided the targets into the development set and the test set. To this aim, 10 targets, namely 'feel', 'hit', 'look', 'phone', 'rise', 'scream', 'shake', 'smell', 'strike', and 'throw', are selected as the development set and 4 targets, namely 'bend', 'follow', 'forget', and 'rush', are used as test set.

Applying our stopping criterion, in each iteration we compute the variance of classifier's confidence score for the selected samples. The simplest method to stop active learning is when variance starts to decrease. Setting this condition, however, it is very likely to stick in local maxima. In other words, it is possible to have small peaks before reaching the highest variability of the classifier's confidence score, the global peak, in which we are not interested in and we should ignore them. To avoid such a problem, by looking at the variance curves of the development set, we found that active learning should stop when variance (V) decreases in two sequential iterations _ we call it model 1; i.e. $V_i < V_{i-1}$ and $V_{i-1} < V_{i-2}$. There is a possibility that this condition is not satisfied at all. In such cases, active learning would not be stopped and all data should be labeled. Fortunately, this situation has happened very rarely in our data such that it happened only 3 times in the development set: 'look' (fold 1), 'hit' (fold 4), and 'bend' (fold 3) and it did not happen in the test targets at all. Having this stopping criterion, we have applied it on each fold of the two active learning sampling methods. Our idea is shown in Figure 19 for fold 5 of the target 'rise', such that the proposed stopping criterion is satisfied in iteration 11:

Figure 19. Variance curve of the verb 'rise'

To evaluate how well our stopping criterion is, we have compared the average performance of classifier on the stopping points of 5 folds with the maximum performance of classifier in the learning curve in which the whole data is labeled. The summary of the result for uncertainty sampling method is shown in Table 6 both for the development set and the test set which are stared.

Table 6. The comparison of average performance of classifier on the stopping point with the maximum performance in uncertainty sampling

Verb	Uncertainty Sampling Performance	
	Stopping point	Maximum
Feel	70.37	73.07
Hit	63.56	65.71
Look	80.00	81.76
Phone	93.54	93.54
Rise	75.45	78.18
Scream	62.85	72.14
Shake	52.38	57.13
Smell	74.81	76.29
Strike	53.00	64.00
Throw	81.38	80.68
* bend	53.00	53.00
* follow	70.00	71.81
* forget	41.00	51.00
* rush	89.03	89.03

Comparing the applied stopping criterion in the test set with the maximum performance of the uncertainty sampling, we can see that in 2 targets (‘bend’, ‘rush’) the maximum performance of the classifier is achieved in the stopping point. In one target (‘follow’) there is a small reduction of performance in which 97.47% of the maximum performance is kept. While for the other one target (‘forget’) there is a big loss in performance in which only 80.39% of the maximum performance is achieved.

Having applied the stopping criterion on over-sampling method, the result is summarized in Table 7. As represented in the table, applying the stopping criterion on the test set, we did not reach the maximum performance in none of the targets. However, in three of the targets (‘bend’, ‘follow’, ‘rush’) more than 95% of the maximum performance is kept; and only in one target (‘forget’) 87.75% of the maximum performance is achieved.

Table 7. The comparison of average performance of classifier on the stopping point with the maximum performance in over-sampling

Verb	**Over-sampling Performance**	
	Stopping point	**Maximum**
Feel	63.84	74.61
Hit	55.71	63.57
Look	77.05	82.94
Phone	93.54	93.54
Rise	72.72	78.18
Scream	62.85	70.00
Shake	56.18	62.85
Smell	74.07	75.55
Strike	57.00	64.00
Throw	75.86	80.00
* bend	49.00	51.00
* follow	70.90	73.63
* forget	43.00	49.00
* rush	88.38	89.03

Looking carefully on the variance curves of 5 folds of the development set, we have seen that in some iteration the decreased variance in two sequential iterations is very small and it is still possible to stick in local maxima as observed in Figure 20 in iteration 8 of fold 3 of the target ‘look’.

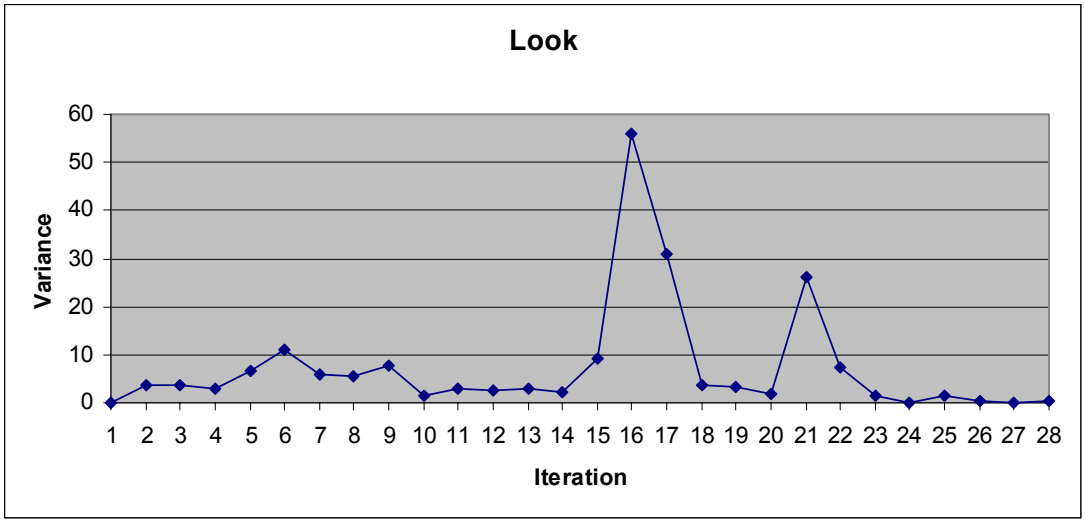

Figure 20. Variance curve of the verb 'look'

To have a better stopping point, we set a threshold in the development set such that the decreasing variance in two sequential iterations must be bigger than 0.5; i.e. $V_i < V_{i-1} - 0.5$ and $V_{i-1} < V_{i-2} - 0.5$ _we call it model 2_ so that in Figure 20 we will stop in iteration 18. We applied this model on test set and compared to model 1. We found out for 1 target ('forget') model 2 has achieved a very good performance, for 2 targets ('follow', 'rush') it was ineffective; and for 1 target ('bend') it caused to have a small reduction in performance.

To show the advantage of model 2 compared to model 1, we have represented the results as a ratio: the ratio is the average performance of the classifier for 5 folds when reaching the stopping point divided by the maximum performance of the classifier when all data annotated. In fact, we calculated the ratio by dividing column two of Table 6 to the column three of the same table; so, the higher ratio the better. The summary of ratios of models 1 and 2 for uncertainty sampling is shown in Table 8. As represented in the table, the average ratio of the test set in model 2 is higher than the average ratio of the test set in model 1. So, we can conclude that based on the results that we have model 2 is a better stopping point for uncertainty sampling.

Table 8. The comparison of two ratios of the stopping points

Verb	Uncertainty Sampling Performance	
	Model 1	**Model 2**
Feel	96.29	97.35
Hit	96.73	96.73
Look	97.84	100
Phone	100	100
Rise	96.51	96.51
Scream	87.12	88.11
Shake	91.67	93.34
Smell	98.06	96.11
Strike	82.81	82.81
Throw	100	99.14
* bend	100	98.11
* follow	97.47	97.47
* forget	80.39	90.19
* rush	99.99	99.99
Average on Development Set	**94.79**	**95.01**
Average on Test Set	**94.46**	**96.44**

Table 9. The comparison of two ratios of the stopping points

Verb	Over-sampling Performance	
	Model 1	**Model 2**
Feel	85.56	90.72
Hit	87.64	87.64
Look	92.90	94.33
Phone	100	100
Rise	93.02	96.51
Scream	89.79	91.83
Shake	89.39	96.97
Smell	98.03	98.03
Strike	89.06	93.75
Throw	94.82	93.96
* bend	96.07	96.07
* follow	96.29	96.29
* forget	87.75	87.75
* rush	99.27	99.99
Average on Development Set	**92.02**	**94.37**
Average on Test Set	**94.85**	**95.03**

Could model 2 be used for over-sampling too? We have applied the two models for over-sampling method and represented the results as ratios in Table 9. Again, the

average ratio of the test set in model 2 is higher than the average ratio of the test set in model 1 and we can conclude model 2 is a better stopping criterion for over-sampling as well.

From all the results of model 1 and 2 for uncertainty sampling and over-sampling we generalize our finding on the stopping point which indicates model 2 is better for the two active learning sampling methods based on the results that we have.

6.4 Discussion

Above we presented the results for the targets that indicated active learning performs better than random sampling. Here the question that was raised in Section 3.5 should be repeated again that 'does active learning could always be a help?' It is hard to say strongly 'yes' as Baldridge and Osborne (2004) and Dang (2004) reported active learning did not performed better than random sampling in their experiments.

In our study, we also achieved results on some of the targets that show active learning did not perform better than random sampling. The results of two targets, namely 'forget' and 'scream', are presented in Figures 21 and 22.

As presented in the graphs, random sampling outperformed active learning. We believe there is a reason behind it to have such kinds of results. Looking carefully at the data and the features that the classifier has used could be the guideline.

What we did was finding some data that have a lot of common features between different labels. We have called such data 'uncertain data' which is very difficult for the machine to label. Asking oracle to label such samples, even though the oracle is totally uncertain about the labels, he has to assign only one label to them. Assigning a single label to such uncertain data could be a source of problem to the classifier since the classifier is trained with features of this data and it is misleaded; because the classifier would make wrong decisions on the labels of the new unlabeled data or test data which have similar features with the uncertain data. In other words, we belive having a large amount of overlapping features in features sets of different frames misleads the classifier.

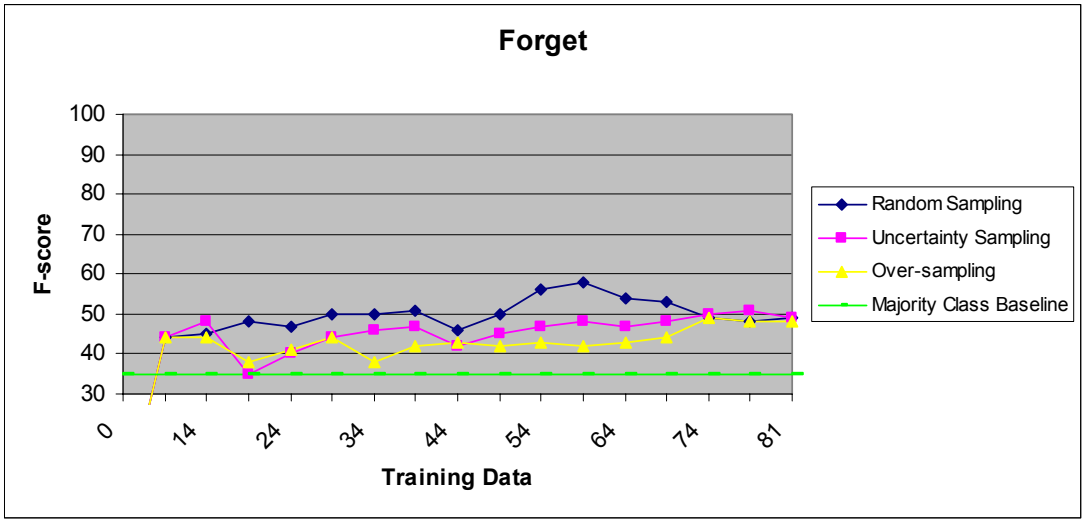

Figure 21. Learning curve of the verb 'forget'

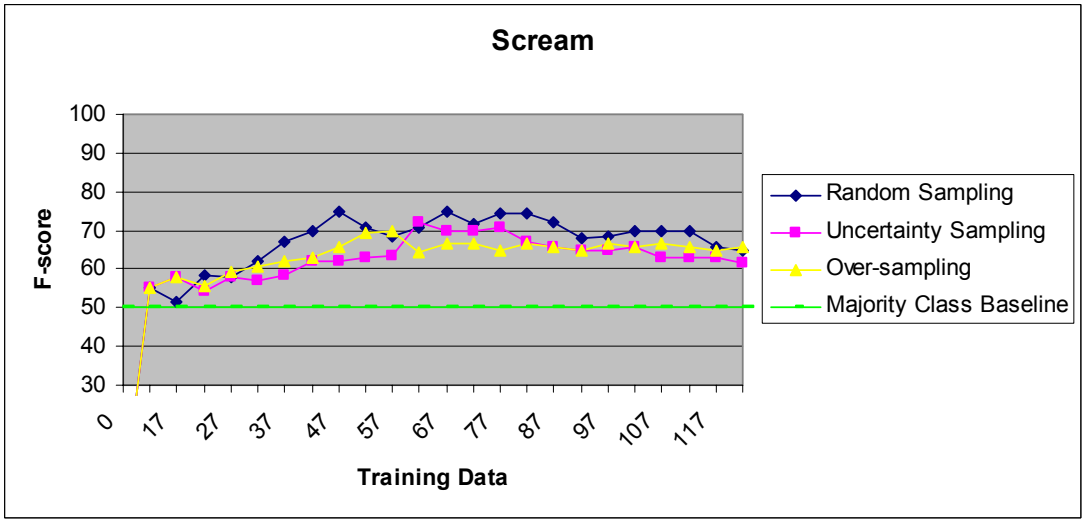

Figure 22. Learning curve of the verb 'scream'

Looking at the samples of the targets 'forget' and 'scream', it is so frequent to have the overlapped features in data. If the classifier is trianed with more data which does not have a lot of overlapped features, there is a possibility to resolve the problem of labeling instances having overlapped feaures by giving appropriate weights to the features. However, since we suffer from the sparsity problem, the problem of overlapping features remains and the classifier can not deal with such noisy data; so, it performs poorly.

To be more precise on this point, we here show some samples of the target 'forget' in which there is a large portion of overlapping in their features. Presenting particular

instances do not mean we want to make any conclusions out of them; but only to make our point clear what the characteristics of uncertain samples are and how they effect on the classifier. Considering this, we have brought up these three instances. 'forget' is a verb having three frames, namely 'REMEMBERING_EXPERIENCE', 'REMEMBERING_INFORMATION', and 'REMEMBERING_TO_DO' (having 27 samples as extra unlabeled data for each frame and 7, 6, and 7 samples as test data respectively). As could be seen in Figure 21, active learning suddenly drops in iteration 3 and it requires more data to reach the same level of performance, which is achieved in iteration 11; and as shown random sampling has almost incremental performance. Checking the extra unlabeled data, we found three instances which have overlapped features as they are given below. In front of each of them, the features that are extracted, the names of that utilized plugin are written*.

1.
```
TARGET: Do n't forget your bag . ‘
FRAME:  Remembering_to_do
WordRangeClusterer [1.0 [StanfordPosTagger]]
       n't Do  your bag .
StanfordPosTagWordRangeClusterer [1.0 [StanfordPosTagger]]
       RB VBP   NN
SentencePhraseClusterer [1.1 [BerkeleyParser]]
       forget your bag
SentencePhrasePosTagClusterer [1.1 [BerkeleyParser]]
       VB Unknown NN
```

2.
```
TARGET: ' I have n't forgotten yesterday . ‘
FRAME:  Remembering_experience
WordRangeClusterer [1.0 [StanfordPosTagger]]
       n't have I yesterday . ''
StanfordPosTagWordRangeClusterer [1.0 [StanfordPosTagger]]
       RB VBP  NN
SentencePhraseClusterer [1.1 [BerkeleyParser]]
       have n't forgotten yesterday
SentencePhrasePosTagClusterer [1.1 [BerkeleyParser]]
       VBP RB VBN NN
```

3.
```
TARGET: We paused now and again at some hostelry and , on one
occasion , a Benedictine monastery , I forget its name .
FRAME:  Remembering_information
WordRangeClusterer [1.0 [StanfordPosTagger]]
       I , monastery its name .
StanfordPosTagWordRangeClusterer [1.0 [StanfordPosTagger]]
         NN  NN
SentencePhraseClusterer [1.1 [BerkeleyParser]]
       forget its name
SentencePhrasePosTagClusterer [1.1 [BerkeleyParser]]
       VB Unknown NN
```

* As observed, pronouns are not tagged due to having a bug in SLSP tool. We have done new experiments with the debugged tool for two targets as discussed in Section 6.5.

To find the overlapped features, we have taken a systematic procedure: since we have utilized different plugins in SLSP, they should be treated separately; as a result, we find the overlapped features of each plugin independent to other plugins; since the window size ±3 is defined for two of the plugins (Stanford Word Range Plugin and Stanford POS Tag Word Range Plugin), the positions of the features are so important and the overlapped features with respect to their positions should be considered. As SLSP tool determines 14 different features[*], they are compared separately. These 14 features include 6 different positions for Stanford Word Range Plugin, 6 different positions for Stanford POS Tagger Plugin, 1 position for Berkley Sentence Phrase Plugin, and 1 position for Berkley Sentence Phrase POS Tag Plugin. Let's see what happens if the system is trained with such samples and the classifier uses the features out of them. We have done the experiments both on uncertainty sampling and over-sampling methods. At first we train the system with seed data which was not overlapped with the above 3 instances. It should be pointed out that the seed size of 'forget' is 9 such that 3 instances from each frame are available. Before adding any sampling to the training data, we asked the classifier to predict the frames of these three instances as we gave them to the system unlabeled. The system predicted the frames correctly but with a low confidence:

```
REMEMBERING_TO_DO [35%] Do n't forget your bag . ‘
REMEMBERING_EXPERIENCE [38%] ' I have n't forgotten yesterday . ‘
REMEMBERING_INFORMATION [45%] We paused now and again at some
hostelry and , on one occasion , a Benedictine monastery , I forget
its name .
```

As shown, instance 1 has the lowest confidence. So, it is one potential candidate that is selected by active learner. To see the behavior of a sample which has overlapped features on the classifier, we added instance 1 to the training data and retrained the

[*] Recalling from Section 5.3.1, in SLSP features are indexed as C*i*. For plugin 1 the features are C0 for position -3, C1 for position -2, C2 for position -1, C3 for position +1, C4 for position +2, and C5 for position +3. For plugin 2 the features are C6 for position -3, C7 for position -2, C8 for position -1, C9 for position +1, C10 for position +2, and C11 for position +3. For plugin 3 and plugin 4 all features are C12 and C13, respectively, which indicates the position and order is not important for the features of these plugins.

system while considered the other two instances as test data. Then, we asked the system to predict the frames of the other two instances. The predicted frames were surprising because the classifier assigned the label ‘REMEMBERING_TO_DO’ to both samples:

```
REMEMBERING_TO_DO [52%] ' I have n't forgotten yesterday . ‘
REMEMBERING_TO_DO [60%] We paused now and again at some hostelry and
, on one occasion , a Benedictine monastery , I forget its name .
```

As it could be observed, when an uncertain sample such as instance 1 is picked up by active learning in early iterations because of its low confidence, it makes the classifier misleaded and the labels that have already predicted correctly are now predicted wrongly. It could be concluded that training the system with such uncertain samples has an impact on the classifier throughout the training process and causes to drop the performance.

Another question might be raised is that when we only added instance 1 to the training data, we have imposed the imbalanced problem to the classifier; so this label has the higher prior probability than the other frames. To check this, we have retrained the system with over-sampled data to resolve the imbalanced problem. Again we predicted the frames of instances 2 and 3. The prediction of the classifier did not change and it was still ‘REMEMBERING_TO_DO’ for both samples; but with a little decrease on its confidence:

```
REMEMBERING_TO_DO [46%] ' I have n't forgotten yesterday . ‘
REMEMBERING_TO_DO [53%] We paused now and again at some hostelry and
, on one occasion , a Benedictine monastery , I forget its name .
```

Based on what described, we conclude that such uncertain samples can mislead the classifier easily. Since active learning selects the most uncertain samples in each iteration, such uncertain data will be selected in early iterations; as a result, the classifier will malperform; while in random sampling there is a possibility to have such samples in next iterations and till then the classifier has learnt enough and could

have a high confidence score on the label of such samples so that they will not mislead the classifier easily.

To quantify this finding, we hypothesized when there is a high degree of overlapping on features, it is possible that the classifier does not perform properly. To this aim, we selected two targets: ‘forget’ which has 3 frames and has a bad performance of active learning; and ‘smell’ which has the same number of frames but a good performance of active learning. We have tried to find the overlapped features of different plugins for 30 sentences selected in the 6 mid iterations of the two targets which are more interesting for us. Summing up the number of overlapped features between different frames, we found out ‘forget’ has more overlapped features than ‘smell’; so regarding the two targets, our hypothesis for poor performance of active learning when there is a large amount of overlapped features will be true. The detailed numbers of overlapped features with respect to the utilized plugins (Plugin 1 to Plugin 4) and their features (C0 to C13) among different pairs of frames for targets ‘forget’ and ‘smell’ which are compared are given in Table 10.

To know what the most frequent overlapped features and their values are, we have found every individual features and their values that are shared among different pairs of frames for the two targets. 5 frequent values of features for each plugin that has been used in SLSP are presented in Table 11 for the target ‘forget’ and in Table 12 for the target ‘smell’.

Table 10. the number of overlapping features for different frames of targets ‘forget’ and ‘smell’

Target		Forget			Sum	Smell			Sum
Frame 1		REMEMBERING_ INFORMATION	REMEMBERING_ EXPERIENCE	REMEMBERING_ EXPERIENCE		PERCEPTION_ ACTIVE	APPEARANCE	APPEARANCE	
Frame 2		REMEMBERING_ TO_DO	REMEMBERING_ TO_DO	REMEMBERING_ INFORMATION		PERCEPTION_ EXPERIENCE	PERCEPTION_ EXPERIENCE	PERCEPTION_ ACTIVE	
Plugin 1	**C0**	19	30	15	64	7	9	9	25
	C1	2	4	20	26	3	3	2	8
	C2	1	9	1	11	2	3	0	5
	C3	14	11	15	40	2	13	60	75
	C4	7	8	4	19	6	10	6	22
	C5	1	13	2	16	6	12	11	29
Plugin 2	**C6**	77	96	67	240	9	10	1	20
	C7	15	21	28	64	18	18	3	39
	C8	9	7	13	29	1	17	1	19
	C9	41	67	51	159	28	96	99	223
	C10	28	24	16	68	51	107	77	235
	C11	66	116	86	268	47	122	63	232
Plugin 3	**C12**	485	812	744	2041	315	658	445	1418
Plugin 4	**C13**	3137	4403	3988	11528	2728	5608	2374	10710

Table 11. The frequent overlapping features for different frames of targets ‘forget’

Frame 1	Frame 2	Plugin 1		Plugin 2		Plugin 3		Plugin 4	
		Feature	**Freq**	**Feature**	**Freq**	**Feature**	**Freq**	**Feature**	**Freq**
REMEMBERING_ EXPERIENCE	REMEMBERING_ INFORMATION	C0=had	13	C9=RB	60	C12=the	130	C13= Unknown	819
		C3=the	10	C9=IN	19	C12= forgotten	126	C13=NN	764
		C4=the	5	C9=DT	14	C12= forget	37	C13=IN	592
		C0=n’t	4	C6=VBD	14	C12=had	37	C13=RB	410
		C3=by	3	C10=NN	13	C12=a	22	C13=VBN	409
REMEMBERING_ EXPERIENCE	REMEMBERING_ TO_DO	C0=n’t	14	C6=RB	59	C12=the	161	C13= Unknown	1567
		C0=had	9	C9=IN	50	C12= forgotten	158	C13=IN	1110
		C4=the	6	C6=VBD	12	C12=to	118	C13=NN	873
		C3=the	3	C10=NN	7	C12=had	72	C13=VBD	574
		C0=have	3	C9=DT	7	C12=of	37	C13=DT	486
REMEMBERING_ INFORMATION	REMEMBERING_ TO_DO	C1=Do	8	C6=RB	47	C12=the	141	C13= Unknown	841
		C3=the	8	C9=IN	24	C12= forgotten	86	C13=NN	644
		C3=to	6	C7=VBP	20	C12=to	82	C13=IN	594
		C0=had	5	C9=DT	16	C12=had	35	C13=VBD	561
		C1=he	5	C6=VBD	13	C12= forget	34	C13=DT	327

Table 12. The frequent overlapping features for different frames of targets ‘smell’

Frame 1	Frame 2	Plugin 1		Plugin 2		Plugin 3		Plugin 4	
		Feature	**Freq**	**Feature**	**Freq**	**Feature**	**Freq**	**Feature**	**Freq**
APPEARANCE	PERCEPTION_ ACTIVE	C4=and	5	C10=NN	35	C12=the	145	C13=NN	1406
		C0=He	3	C9=DT	18	C12=and	54	C13=IN	621
		C1=a	2	C7=NN	8	C12=a	42	C13=DT	475
		C0=and	2	C7=DT	8	C12=,	38	C13=VBD	254
		C0=she	2	C6=NN	6	C12= smelled	24	C13= Unknown	216
APPEARANCE	PERCEPTION_ EXPERIENCE	C3=of	6	C10=NN	44	C12=,	311	C13=NN	2764
		C4=and	5	C9=DT	39	C12=the	278	C13=IN	2041
		C3=the	3	C10=IN	20	C12=and	123	C13= Unknown	1684
		C3=a	3	C9=JJ	17	C12=of	94	C13=DT	1012
		C0=,	3	C9=IN	14	C12=a	68	C13=JJ	668
PERCEPTION_ ACTIVE	PERCEPTION_ EXPERIENCE	C3=the	51	C9=DT	59	C12=the	246	C13=NN	1094
		C0=I	9	C10=NN	36	C12=,	35	C13=IN	493
		C4=and	2	C9=NN	9	C12=and	29	C13=DT	488
		C3=that	2	C10=CC	2	C12= smell	24	C13= Unknown	235
		C3=a	2	C7=NN	2	C12= smelt	17	C13=VBD	230

6.5 Problems and Bugs

Having done the experiments on 14 targets, we have realized that subjective and objective personal pronouns as well as possessive pronouns are missed and they are not tagged by the plugins in the SLSP tool as it could be seen in the instances on page 63; so that such information as features is not available to the classifier. Missing this type of information beside the data sparseness adds to the problems and it makes it hard to judge about the effect of the bug in the performance since data sparseness makes relatively a large variability in the results.

After applying the debugged system, we have done all experiments of sampling methods in 5 folds for only 2 targets, namely ‘forget’ and ‘shake’, and then tried to compare the results of the old system (Figures 23 and 25) with the debugged system

(Figures 24 and 26). Training with more features, we expected to have changes in the performance and the final results; but the changes were not huge as we compared the learning curves of the targets. 'shake', which had a good performance with old feature set, shows a decrease in performance when the new feature set is used. While for the target 'forget' it is vise versa: active learning that had a very poor performance with old features, it still has kept the poor performance with new feature but in mid iterations it has a better performance than random sampling. The reason that we could think of is that for the target 'shake' the new features are not needed; so, they are not useful; while for the target 'forget' the new features are needed as they provide more information to the classifier.

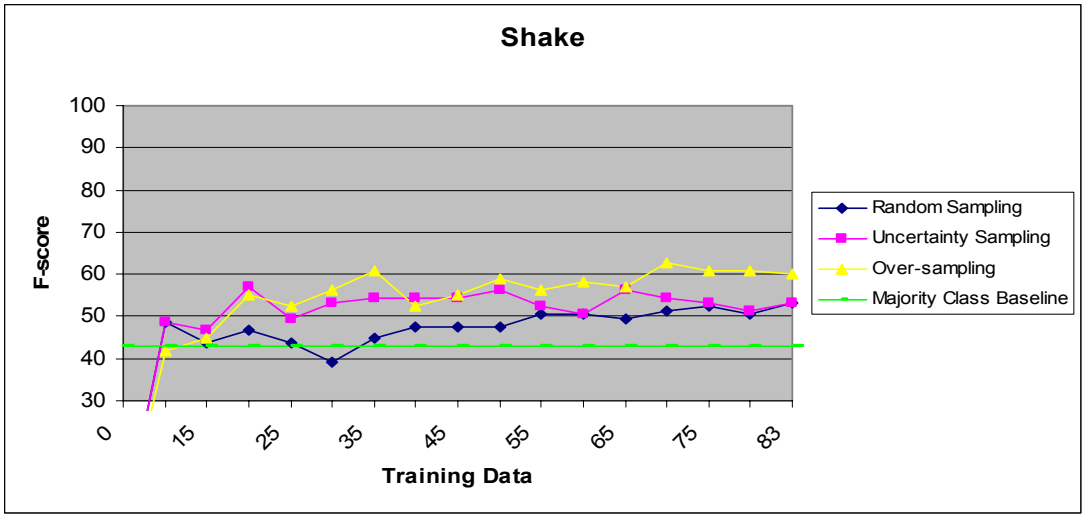

Figure 23. Learning curve of the verb 'shake' for the old system

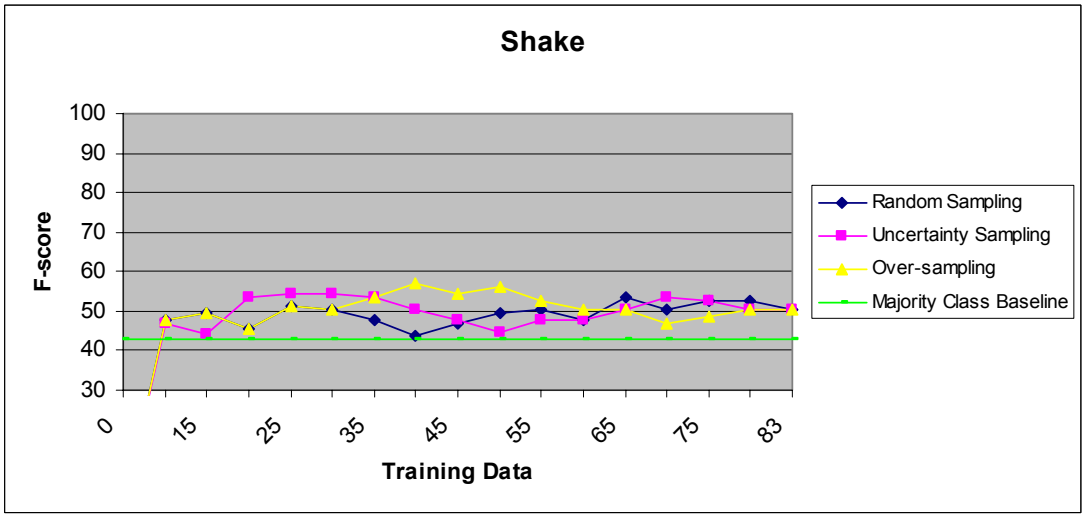

Figure 24. Learning curve of the verb 'shake' for the debugged system

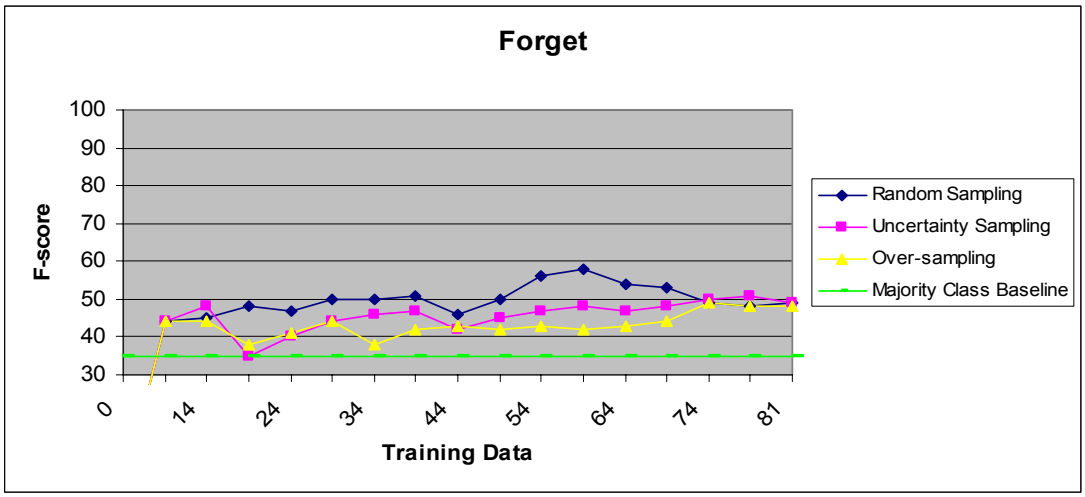

Figure 25. Learning curve of the verb 'forget' for the old system

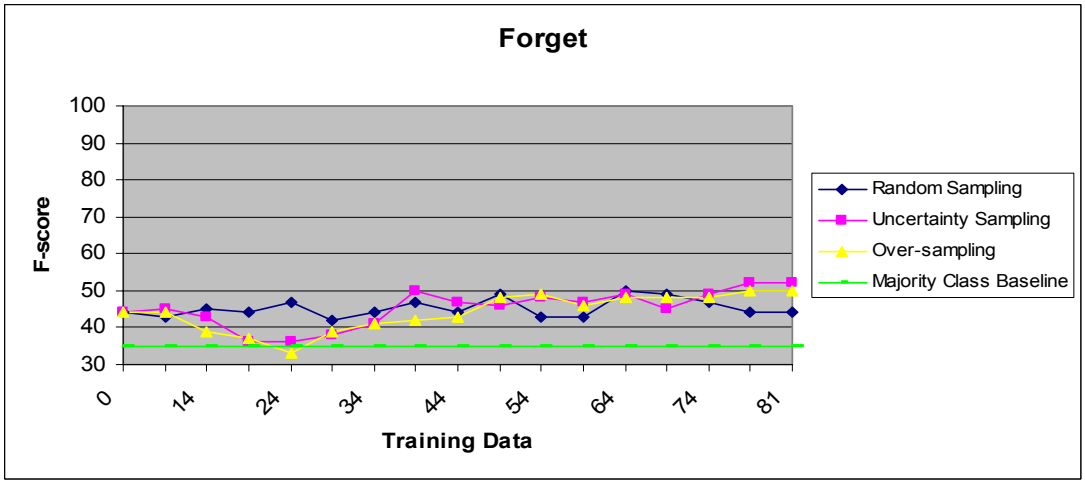

Figure 26. Learning curve of the verb 'forget' for the debugged system

6.6 Summary

This chapter devoted to the results we got from our experiments for frame assignment with the help of active learning for three different sampling methods. We found three types of behavior for active learning. For 7 targets the performance of active learning was better than the baseline; over-sampling which was used to resolve the imbalance problem had a better performance in two target compared to uncertainty sampling and a worse performance in other targets. The possible reasons we thought of was having a high degree of over-sampling that has made the training data overfitted and as a result the classifier malperformed; or over-sampling the minority class has made the classifier limited to the extracted features from these

samples and if the features of the test data differed from the features of the extra data, then the classifier malperformed.

For 5 targets active learning was not that much effective. One possible reason was related to the number of frames and the available data to each frame. While by having a high degree of skewness, none of the baselines could beat the majority class baseline. We also discussed for 2 targets active learning had a poor performance compared to the random sampling as the baseline. The reason we thought of was having a large amount of features shared among different frames in the unlabeled data which was a source of problem for the classifier and to label such samples.

The other issue we discussed was the near-optimum stopping point for uncertainty sampling and over-sampling. To this aim, we proposed two models based on the variability of classifier's confidence score on the instances selected in each iteration of different folds. The proposed two models kept more than 90% of maximum performance of the classifier while model 2 had a slight improvement in the performance compared to model 1.

Chapter 7

Summary, Conclusion, and Future Work

7.1 Summary and Conclusion

In our study we focused on frame semantics structure analysis as one of the natural language understanding techniques such that in this analysis, the semantic roles of elements participated in the action should have been identified. To determine the roles automatically, two steps are required: one is frame assignment, and the other one is role assignment. Frame assignment plays a more significant role than role assignment since the appropriate frame elements depend on the assigned frame.

What we aim to do is assigning frames with a supervised machine learning method called 'active learning'. Supervised learning methods require a huge amount of labeled data. Since labeling the data is very difficult, time consuming, error prone, costly, and expensive to obtain, active learning promises to maximize the performance by minimizing the human's effort in labeling the data.

Among active learning scenarios and sampling methods, we have selected pool-based active learning with uncertainty sampling method. We have chosen 14 frequent targets from FrameNet data set for our task. Random sampling which represents the distribution of frames in the corpus would be our baseline to find how effective active learning is. Since for each target there was at least one dominant

frame, we face the imbalance problem which might have a negative impact on the classifier; so over-sampling is used to resolve this problem.

According to the results, active learning worked out for most of the target; for some of them it was not that much effective; and for some it performed poorly and could not be a help. We have discussed this issue in details and we found out having a huge amount of overlapped features for data in different frames might mislead the classifier and since active learning selects the informative samples, the selection of uncertain samples makes problem for the classifier in order to annotate the new data. Regarding the results we have, it is very difficult to predict the behavior of the final results for new targets and classify them to any of the 3 types of behavior. As a result, the data and the features available to the data are very important properties that should be considered.

As we know active learning is an iterative process, it should be stopped at a point when the classifier has reached to its maximum performance. Reaching this point is so difficult; so, we have proposed a new stopping criterion which stops active learning in a near-optimum point. This stopping criterion is based on the confidence score of the classifier on the extra unlabeled data such that it uses the variance of the classifier's confidence score for a certain number of samples which are selected in each iteration. As variance graph represented, the classifier has three behaviors: untrained, training, and trained. We believe that when variance has reached to its global maxima and it starts to decrease is a good point to stop; in other words, the stopping point is when the classifier passes from the training step to the trained step. To avoid local maxima, we suggested when variance decreases in two sequential iterations, then it is a good stopping point. We have modified the stopping point to have a better performance such that we should stop when variance decreases in two sequential iterations and the decrement is more than 0.5.

7.2 Future Work

There are different issues that could be considered as the future work. One possibility is choosing a different query method for frame assignment such as query-by-committee, or ignoring to choose outliers in uncertainty sampling. It would be interesting if the effect of active learning for role assignment be studied too. The other possibility is improving our proposed stopping criterion. In the current stopping criterion, we have used the variance of classifier's confidence score only for selected samples in each iteration of different folds. It would be interesting if the variability of classifier's confidence scores for all training data in each iteration be considered and see whether we can reach a better point to stop.

References

Baker, C.F. and C.J. Fillmore and J.B. Lowe (1998) 'The Berkeley FrameNet project' In *Proceedings of the joint Annual Meeting of the Association for Computational Linguistics and International Conference on Computational Linguistics*, Montreal, QC, pp. 86-90.

Baldridge, J. and M. Osborne (2003) 'Active learning for HPSG parse selection' In *Proceedings of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, pp. 17-24.

Baldridge, J. and M. Osborne (2004) 'Active learning and the total cost of annotation' In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Forum Convention Center, Barcelona, Spain, pp. 9-16

Baldridge, J. and M. Osborne (2006) 'Active learning and logarithmic opinion pools for HPSG parse selection' In *Natural Language Engineering*. 14(2): 199-222. Cambridge, UK.

Baram, Y. and R. El-Yaniv and K. Luz (2004) 'Online choice of active learning algorithm' In *Journal of Machine Learning Research* 5, pp. 255-259

Becker, M. and M. Osborne (2005) 'A two-stage method for active learning of statistical grammars' In *Proceedings of International Joint Conferences on Artificial Intelligence,* Edinburgh, pp. 991-996.

Burchardt, A. and K. Erk and A. Frank and A. Kowalski and S. Padó and M. Pinkal (2006) "SALTO: A versatile multi-level annotation tool" In *Proceedings of LREC 2006*, Genoa, Italy.

Busser, B. and R. Morante (2005) 'Designing an active learning based system for corpus annotation' In *Procesamiento del Lenguaje Natural*, núm. 35, pp. 375-381.

Carreras, X. and L. Márquez (2004) Introduction to the CoNLL-2004 shared task: Semantic role labeling. In *Proceedings of the CoNLL Shared Task*, Boston MA, USA.

Carreras, X. and L. Márquez, (2005) Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the CoNLL Shared Task*, Ann Arbor, Michigan, USA.

Chan, Y.S. and H.T. Ng (2007) 'Domain adaptation with active learning for word sense disambiguation' In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, Prague, pp. 49-56.

Chen, J. and A. Schein and L. Ungar and M. Palmer (2006) 'An empirical study of the behavior of active learning for word sense disambiguation' In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, New York. pp. 120-127.

Chklovski, T. and R. Mihacea (2002) "Building a sense tagged corpus with Open Mind Word Expert" In *Proceedings of the ACL Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*.

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

Dagan, I. and S.P. Engelson (1995) 'Committee-based sampling for training probabilistic classifiers' In *Proceedings of the 12th International Conference on Machine Learning*, San Francisco, CA. Morgan Kaufman, pp. 150-157.

Dang, H.T. (2004) *Investigation into the Role of Lexical Semantics in Word Sense Disambiguation*. PhD dissertation, University of Pennsylvania, USA.

Engelson, P. and I. Dagan (1996) 'Minimizing manual annotation cost in supervised training from corpora' In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*

Erk, K. (2005) 'Frame assignment as word sense disambiguation' In *Proceedings of the 6th International Workshop on Computational Semantics*, Tilburg, the Netherlands.

Erk, K. and S. Padó (2006) "SHALMANESER: A toolchain for shallo semantic parsing" In *Proceedings of LREC 2006*, Genoa, Italy

Fillmore, C.J. (1968) 'The case for case.' In Emmon W. Bach and Robert T. Harms, editors, *Universals in Linguistic Theory*. Holt, Rinehart & Winston, New York, pp. 1-88.

Fillmore, C.J. (1971) 'Some problems for case grammar.' In R. J. O'Brien, editor, *22nd Annual Round Table. Linguistics: Developments of the Sixties-Viewpoints of the Seventies*. Volume 24 of *Monograph Series on Language and Linguistics*. Georgetown University Press, Washington, D.C., pp. 35-56.

Fillmore, C.J. (1982) 'Frame Semantics' In *Linguistics in the morning calm*, Seoul, Korea: Hanshin, pp. 111-138.

Fillmore, C.J. (1985) 'Frames and the semantics of understanding' In *Quaderni di Semantica*, 6.2:222-254

Fillmore, C.J. (1994) 'Starting where the dictionaries stop: The challenge of corpus lexicography' In *Computational Approaches to the Lexicon*, ed. By B.T.S. Atkins and A. Zampolli, Oxford, pp. 349-393.

Freund, Y. and H. Seung and E. Shamir and N. Tishby (1997) 'Selective sampling using the query by committee algorithm' *Machine Learning*, 28, pp. 133-168.

Fujii, A. and T. Tokunaga and K. Inui and H. Tanaka (1998) 'Selective sampling for example-based word sense disambiguation' *Computational Linguistics*, 24(4): 573–597.

Gildea, D. and D. Jurafsky (2002) 'Automatic labeling of semantic roles' In *Association for Computational Linguistics*, Vol. 28, Num. 3, pp245-288.

Haffari, G. and A. Sarkar (2009) 'Active learning for multilingual statistical machine translation' In *Proceedings of the 47th annual meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP)*, Singapore.

Haffari, G. and M. Roy and A. Sarkar (2009) 'Active learning for statistical phrase-based machine translation' In *Proceedings of the annual meeting of the North*

American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT). Boulder, Colorado.

Hakkani-Tür, D. and G. Riccardi and A. Gorin (2002) 'Active learning for automatic speech recognition' In *Proceedings of 'International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Orlando, FL.

Hearst, M. (1999) 'Untangling text data mining' In *Proceedings of the 37th Annual Meeting of the ACL*, College Park, Maryland, pp. 3-10.

Hoi, S.C.H. and R. Jin and M.R. Lyu (2006) 'Large-scale text categorization by batch mode active learning' In *Proceedings of the International Conference on the World Wide Web*, pp. 633–642.

Hwa, R. (2000) 'Sample selection for statistical grammar induction' In *Proceedings of the 2000 Joint SIGDAT Conference on EMNLP and VLC*, Hong Kong, China, pp. 45.52.

Hwa, R. and M. Osborne and A. Sarkar and M. Steedman (2003) 'Corrected co-training for statistical parsers' In *Proceedings of the ICML Workshop: The Continuum from Labeled to Unlabeled Data.* pp. 95-102.

Japkowicz, N. (2000) 'Learning from imbalanced data sets: A comparison of various strategies' *In Proceedings of Learning from Imbalanced Data Sets, Papers from the AAAI Workshop, Technical Report WS-00-05 ,* Japkowicz, N. (editor)*,* pp. 10-15.

Laws, F. and H. Schütze (2008) 'Stopping criteria for active learning of named entity recognition' In *Proceedings of the 22nd International Conference on Computational Linguistics (CoLing)*, Manchester, UK, pp. 465-472.

Lewis, D.D. and J. Catlett (1994) 'Heterogeneous Uncertainty Sampling for Supervised Learning' In *Proceedings of the 11th International Conference on Machine Learning*, pp.148-156.

Lewis, D.D. and W. Gale (1994) 'A sequential algorithm for training text classifiers' In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3-12.

Liere, R. and P. Tadepalli (1997) 'Active learning with committees for text categorization' In *Proceedings 14th Conference of the American Association for Artificial Intelligence (AAAI)*, pp. 591-596.

McCallum, A. and K. Nigam (1998) 'Employing EM in pool-based active learning for text classification' In *Proceedings of the 15th International Conference on Machine Learning (ICML)*, Madison, Wisconsin USA.

Mihalcea, R. and P. Edmonds (2004) Introduction to SENSEVAL: Evaluation of Systems for the Semantic Analysis of Text. In *Proceedings of Senseval-3: The Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain.

Ngai, G. and D. Yarowsky (2000) 'Rule writing or annotation: Cost-efficient resource usage for base noun phrase chunking' In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, Hong Kong, pp. 117-125.

Open NLP MAXENT package:
http://maxent.sourceforge.net

Osborne, M. and J. Baldridge (2004) 'Ensemble-based active learning for parse selection' In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, Boston.

Padó, S. (2007) *Cross-lingual Annotation Projection Models for Role-Semantic Information*. PhD dissertation, Saarland University, Saarbücken, Germany.

Rätsch, G. (2004) 'A brief introduction into machine learning' In *21st Chaos Communication Congress*, Berliner Congress Center, Berlin, Germany.
http://www.ccc.de/congress/2004/fahrplan/files/105-machine-learning-paper.pdf

Ruppenhofer, J. and M. Ellsworth and M. R.L.Petruck and C. R.Johnson and J. Scheffczyk (2006) *FrameNet II: Extended Theory and Practice*
http://framenet.icsi.berkeley.edu/

Roth, D. and K. Small (2006) 'Margin-based active learning for structured output space' In *Proceedings of the European Conference on Machine Learning (ECML)*.

Roy, N. and A. McCallum (2001) 'Toward optimal active learning through sampling estimation of error reduction' In *Proceedings of 18th International Conference on Machine Learning (ICML)*, pp. 441-448.

Sassano, M. (2002) 'An empirical study of active learning with support vector machines for Japanese word segmentation' In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, PA.

Schohn, G. and D. Cohn (2000) 'Less is more: Active learning with support vector machines' In *Proceedings of 17th International Conference on Machine Learning*, Stanford University.

Settles, B. (2008) *Curious Machines: Active Learning with Structured Instances*. Ph.D. dissertation, University of Wisconsin–Madison, USA.

Settles, B. (2009) 'Active learning literature survey' *Computer Sciences Technical Report 1648*, University of Wisconsin–Madison.

Settles, B. 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 (EMNLP)*, pp. 1069–1078.

Seung, H.S. and M. Opper and H. Sompolinsky (1992) 'Query by committee' In *Computational Learning Theory*, pp. 287-294.

Shannon, C. E. (1948) 'A mathematical theory of communication' *Bell System Technical Journal*, 27:379-423,623-656.

Shen, D. and J. Zhang and J. Su and G. Zhou and C. Tan (2004) 'Multi-criteria-based active learning for named entity recognition' In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL)*, Barcelona, Spain, pp. 589-596.

Steedman, M. and R. Hwax and S. Clark and M. Osborne and A. Sarkar and J. Hockenmaier and P. Ruhleny and S. Bakerz and J. Crimy (2003) 'Example selection for bootstrapping statistical parsers' In *Proceedings of the Human Language Technology Conference / North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*, Edmonton, Canada.

Tang, M. and X. Luo and S. Roukos (2002) 'Active learning for statistical natural language parsing' In *Proceedings of the 40th Annual Meeting of the ACL*, Philadelphia, Pennsylvania, USA, pp. 120-127.

Thompson, C.A. and M.E. Califf and R.J. Mooney (1999) 'Active learning for natural language parsing and information extraction' In *Proceedings of the 16th International Conference on Machine Learning*, pp. 406-414.

Thompson, C. and R. Levy and C. Manning (2003) 'A generative mode for FrameNet semantic role labeling' In *Proceedings of the 14th ECML*, Croatia.

Tong, S. (2001) *Active Learning: Theory and Applications*. Ph.D. dissertation, Stanford University.

Tong, S. and D. Koller (2001) 'Support vector machine active learning with applications to text classification' *Journal of Machine Learning Research*, 2:45-66.

Tur, G. and D. Hakkani-Tür and R.E. Schapire (2005) 'Combining active and semi-supervised learning for spoken language understanding' *Speech Communication*, 45(2):171–186.

Tur, G. and R.E. Schapire and D. Hakkani-Tür (2003) 'Active learning for spoken language understanding' In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Hong Kong.

Vlachos, A. (2008) 'A stopping criterion for active learning' In *Journal of Computer, Speech and Language*, Volume 22, Issue 3, pp. 295-312.

Yu, H. (2005) 'Selective sampling for ranking with application to data retrieval' In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 354-363.

Zhang, C. and T. Chen (2002) 'An active learning framework for content-based information retrieval' *IEEE Transactions on Multimedia*, 4(2):260-268.

Zhou, Z. and X. Liu (2006) 'Training cost-sensitive neural networks with methods addressing the class imbalance problem' *IEEE Transactions on Knowledge and Data Engineering*, 18(1):63-77

Zhu, J. and E. Hovy (2007) 'Active learning for word sense disambiguation with methods for addressing the class imbalance problem' In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Prague. pp. 783-790.

Zhu, J. and H. Wang and E. Hovy (2008a) 'Learning a stopping criterion for active learning for word sense disambiguation and text classification' In *Proceedings of the 3rd International Joint Conference on NLP (IJNLP)*, Heydarabad, India. pp. 366-372.

Zhu, J. and H. Wang and T. Yao and B. Tsou (2008b) 'Active learning with sampling by uncertainty and density for word sense disambiguation and text classification' In *Proceedings of the 22nd International Conference on Computational Linguistics (CoLing)* pp. 1137-1144.

Zhu, X. and J. Lafferty and Z. Ghahramani (2003) 'Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions' In *Proceedings of the ICML Workshop on the Continuum from Labe eled to Unlabeled Data*, pp. 58–65.