CALIBRATION ANALYSIS FOR NATURAL LANGUAGE PROCESSING SYSTEMS

An Honor Thesis

Presented by

KHANH X. NGUYEN

Approved by:

___

Brendan O'Connor, School of Computer Science

___

Erik Learned-Miller, School of Computer Science

ABSTRACT

Title: **Calibration analysis for natural language processing systems**

Author: **Khanh X. Nguyen**

Thesis/Project Type: **Project**

Approved By: **Brendan O'Connor, School of Computer Science**

Approved By: **Erik Learned-Miller, School of Computer Science**

We argue that measuring the reliability of probabilistic predictions is crucial for mitigating cascaded errors and estimating risks in natural language processing systems. Hence, we propose *calibration analysis* as a new tool for assessing the probabilistic predictions. Two methods are offered to present analysis results: calibration score and calibration curve. We show that they can be effectively estimated using a simple non-parametric regression method. Moreover, confidence intervals of the estimates can also be constructed. We conduct calibration on different families of probabilistic models and discovered patterns on the effects of model structure, sample size, and feature on calibration.

CHAPTER 1

INTRODUCTION

The ambiguity of natural languages motivates the use of the probabilistic models. Typically, a probabilistic model produces a posterior distribution over the space of all possible output labels for the input. The probability of an output label can be interpreted as the degree of belief that the model holds for the occurrence of that possibility. The model then makes its final decisions based on a decision-making scheme. In this setting, the posterior distribution output by the model is unknown to the users. In addition, performance metrics such as accuracy score or F-1 score also solely measure how to the final decisions align with the true labels and neglected the quality of the predicted posterior distribution.

Many NLP systems use maximum-a-posteriori (MAP), i.e. choosing the most likely outcome, as their decision-making scheme. In complex NLP pipelines, however, MAP inference seems to be inappropriate since downstream models suffer from errors accumulated from upstream models. Using the K-top-most-likely predictions was suggested and shown to yield better performance for several NLP tasks (Sutton and McCallum, 2005; Wellner et al., 2004). Finkel et al. (2006) proposed a more efficient alternative to K-best list, using Monte Carlo sampling to an approximate posterior distributions of upstream models and passed the samples as input for downstream models. Those approaches do not examine the quality of the posterior distributions and assume that they are reliable. This is problematic in practice since most NLP models are imperfect. As a result, the full potentials of those approaches would not have been recognized.

Knowledge about the uncertainty of predictions is also useful for users of NLP systems. Communicating the uncertainty of a model with its users is essential for calculating risk. Consider two predictions for a binary variable: one gives a confidence score of 0.51 that the variable is 1 and the other assigns a score of 0.8 for the same event. Using a MAP decision-making scheme with a threshold of 0.5, the user will receive identical final predictions reporting that the value for the variable is 1 and will trust them equally. However, if they knew the true confidence scores, we would expect that they would trust the former predictions less than the latter one. This type of situation is very common for business users of NLP systems. Prediction confidence score are vital for them in calculating long-term revenue and making investment decisions.

In this thesis, we propose *calibration analysis* as a procedure for measuring the quality of the posterior predictions of probabilistic models. Calibration analysis is a powerful assessment tool for three reasons. First, perfect calibration is one of two main conditions for being perfect at probabilistic prediction. Second, calibration analysis is simple to conduct and the results can be easily communicated to the users through a numerical score or a visualizable plot. Third, calibration analysis is extremely flexible. It is model-independent and application-independent. It can be applied to various types of output depending on the application of interest as long as we have a mechanism to obtain confidence scores from the model. The contributions of this work are:

1. Review the theoretical foundation for calibration analysis for NLP probabilistic models.

2. Describe adaptive binning as a simple nonparametric regression method for calibration analysis. Also, present two means of reporting results of the analysis: calibration scoring and calibration curve plotting.

3. Apply calibration analysis to investigating calibration patterns of models for POS tagging and coreference resolution. Especially, for coreference resolution,

demonstrate a Monte Carlo sampling technique for approximate the confidence scores, which are complicated to compute exactly.

CHAPTER 2

SUMMARY OF WORK OF PREVIOUS RESEARCHERS

Issues on neglecting uncertainty propagation in pipelines that employ MAP scheme was known for many years (Draper, 1995). Efforts to abandon a pipeline architecture in NLP were either increasing the model complexity (Singh et al., 2013; Durrett and Klein, 2014) or ineffective (Sutton and McCallum, 2005). Besides that, approaches that preserves the pipeline and manages to mitigate cascaded errors such as modifying inference algorithms to obtain K best predictions give useful improvements (Huang and Chiang, 2005; Toutanova et al., 2005). Finkel et al. (2006) proposed a more clever way to obtain the top predictions using Monte Carlo inference. Unfortunately, the posterior predictions of the model are assumed to be reliable, which is not always the case for most NLP models. Therefore, calibration analysis will be helpful to better evaluate the effectiveness of those approaches.

The notion of calibration was originally developed in the field of meteorology (Miller, 1962; Murphy, 1973) and was referred to as validity or reliability. Rubin (1984) argued that an applied statistician "should be Bayesian in principle and calibrated to the real world in practice". Murphy and Winkler (1984) proposed a general framework for forecast verification. They proposed two frameworks for analyzing posterior predictions, based on two ways to factorize the joint distribution of prediction and observation. We employ their calibration-refinement framework with a special focus on developing a procedure for measuring miscalibration of NLP models. DeGroot and Fienberg (1983) presented a well-defined theoretical scoring rule for comparing calibration and refinement of forecasters. However, since calculating the score requires knowledge of an unknown realistic distribution of the confidence scores, approximation methods from data sample must be used. In fact, this is a regression problem in which the data points are prediction-observation pairs. Parametric regression models are not flexible enough to generalize calibration patterns of different models. On the other hand, non-parametric models such as local regression (Wasserman, 2006) not only gives better estimates but also provides good confidence intervals of the estimates. Adaptive binning is an easily implemented non-parametric method that is proved to be useful in biochemisty and computer vision (Davis et al., 2007; Leow and Li, 2004). Although we favor this method for the purpose of simplicity, more advanced methods are available such as local likelihood (Frölich, 2006).

Assessing uncertainty of probabilistic predictions is standard in many fields such as weather forecasting (Murphy, 1993), economics (Canova, 1994; Cooley, 1997) or earth sciences (Oreskes et al., 1994) but receives little attention in NLP. Calibration studies for general machine learning models Niculescu-Mizil and Caruana (2005); Caruana and Niculescu-Mizil (2006) showed that applying recalibration techniques boosted performances of various models. Although these works also analyzed calibration and refinements of predictions through visualizable plots, they did not provide a metric to quantify the degrees of miscalibration of the models. Our work not only provides more compact calibration plots but also describe a method to calculate calibration score. Moreover, we also attempt to compute confidence intervals for all of our measurements.

CHAPTER 3

EXPLANATION OF CURRENT METHODOLOGY AND GOALS

3.1 Background

3.1.1 Calibration-refinement framework

Throughout this paper, we will consider a binary prediction problem, where each to-be-predicted instance can either be labeled positive (denoted by 1), or negative (denoted by 0). A probabilistic model for this problem assigns each instance i a *prediction* $q_i \in [0, 1]$, which represents the confidence level that the instance is in the positive class. After the model makes their predictions, the set of true observations will be given for model assessment. The true observation for instance i is denoted by $y_i \in \{0, 1\}$.

Let $S = \{(q_1, y_1), (q_2, y_2), \cdots, (q_N y_N)\}$ be a set of prediction-observation pairs produced by a probabilistic model. Follow Murphy and Winkler (1984), we assume that the elements of S are drawn from a hypothetical joint distribution $P(y, q)$, where y and q are the random variables for the prediction and the true label, respectively. $P(y, q)$ contains all the information needed for analyzing the quality of the predictions. The *calibration-refinement framework* is based on the Bayesian factorization of $P(y, q)$:

$$P(y, q) = P(y \mid q)P(q) \tag{3.1}$$

The conditional probability $P(y = 1 \mid q)$ is called the *realistic frequency* with respect to the prediction value q. It indicates how often the true label turns out to be positive among all instances that are predicted to be positive with a confidence level of q. A model is said to be *perfectly calibrated* (or perfectly reliable) if its predictions match with their realistic frequencies for all confidence levels. A more formal definition of perfect calibration is presented in section 3.1.2. On the other hand, the marginal distribution $P(q)$ reflects a model's refinement. A model is said to be *refined* (or sharp) if $P(q)$ concentrates about 0 and 1. This characteristic indicates that the model is capable of discriminating instances from the positive class from instances from the negative class.

For a more concrete view of the calibration-refinement framework, consider a classic example: *precipitation forecast*. In this task, the forecaster is required to give an assessment on the likelihood of precipitation of each single day in a period of time. If a reliable forecaster give a prediction such as "There is 30% chance that it will rain tomorrow", we should expect that among all the days on which that type of prediction is announced, exactly 30% of them will be rainy. Moreover, we should also expect the same condition to hold for all types of predictions (between 0 and 1). However, a reliable forecaster is not always a "good" predictor. Consider the scenario when a forecaster always predicts the climatological probability, i.e. the long-term frequency of precipitation, for any day. The forecaster will be perfectly calibrated but his or her predictions would be useless for regions where the climatological likelihood of raining and not raining are equally likely. In those cases, such unrefined predictions imply a lot of uncertainty and do not help with finalizing binary decisions.

As we can see, maintaining calibration allows posterior predictions to be more realistic whereas having refinement in predictions reduces uncertainty in the decision-making process. Hence, calibration and refinement are orthogonal and complementary concepts.

3.1.2 Definition of perfect calibration

Consider the set of prediction-observation pairs S defined in the previous section.

Definition 3.1. Given a value q between 0 and 1, inclusively, the *realistic frequency* with respect to q, denoted by p_q, is defined as:

$$p_q = P(y = 1 \mid q) = \lim_{N \to \infty} \frac{\sum_{i=1}^{n} I\{q_i = q\} I\{y_i = 1\}}{\sum_{i=1}^{N} I\{q_i = q\}}$$

where $I\{.\}$ is the indicator function and $(q_i, y_i) \in S$.

Definition 3.2. A set of prediction-observation pairs S is said to be *perfectly calibrated* if and only if:

$$p_q = q \qquad \forall q \in [0, 1].$$

3.1.3 Measuring miscalibration

When a model does fulfill definition 3.2, we say that it is *miscalibrated.* The notion of miscalibration is more interesting to study than perfect calibration since most NLP models fall into this category. It is a natural tempting to devise a metric that quantifies miscalibration. Following DeGroot and Fienberg (1983), we introduce concepts that are necessary for constructing such a metric.

Definition 3.3. Let p be real number in $[0, 1]$. A *strictly proper scoring rule* specified by an increasing function $g_1(x)$ and a decreasing function $g_2(x)$ is a function of x that has the following form:

$$f(x) = p g_1(x) + (1 - p) g_2(x) \tag{3.2}$$

and satisfies that $f(x)$ is maximized only at $x = p$.

Theorem 3.4. If $g_1(x)$ and $g_2(x)$ specify a strictly proper function rule, the overall score S for predictions for a probabilistic predictive model can be expressed in the form $S = S_1 + S_2$, where

$$\begin{aligned} S_1 &= E_q \left[p_q \left(g_1(q) - g_1(p_q) \right) + (1 - p_q) \left(g_2(q) - g_2(p_q) \right) \right] \\ S_2 &= E_q \left[p_q g_1(p_q) + (1 - p_q) \, g_2(p_q) \right] \end{aligned} \tag{3.3}$$

It can be proved that S_1 and S_2 have the following properties:

1. S_1 is zero only for perfectly calibrated model and negative otherwise.
2. If two model A and B are both perfectly calibrated and A is at least as sharp as B, the value of S_2 will be at least as large for A as it is for B.

Choosing $g_1(x) = (x-1)^2$ and $g_2(x) = x^2$, S_1 becomes the expected mean squared error between probabilistic predictions and the corresponding realistic frequencies:

$$CalibMSE = E_q[p_q - q]^2$$

We will refer to calibration score (CalibScore) as the *square root* of CalibMSE.

3.2 Practical calibration analysis

The realistic frequency p_q defined in section 3.1.2 is an unknown quantity. Therefore, the true value of the calibration score cannot be calculated exactly. A general approach for this problem is to replace p_q in the score's formula by an approximation computed from data. We will describe adaptive binning as a simple method for doing it.

Parametric regression is not an appropriate choice since it would not be flexible enough for exploring different models' calibration patterns. Conversely, nonparametric methods only impose weak assumptions on the model choice but still gives close approximation.

3.2.1 Adaptive binning procedure

Adaptive binning is a modified version of *regressogram* (Wasserman, 2006). Instead of dividing the interval $[0, 1]$ into equally spaced like regressogram does, adaptive binning assigns an equal number of data points to each bin. This is advantageous in the context of assessing NLP models, where the distribution of predictions is often skewed toward 0 and 1. Adaptive binning ensures that the mid-range approximations have roughly the same standard errors as those near the boundaries.

Concretely, the adaptive binning procedure is described as follows:

Data: A set of n data points $\{(q_1, y_1), (q_2, y_2), \cdots, (q_N, y_N)\}$.

Parameter: bin size b, the number of points in each bin.

Step 1: Sort the data points by q_i in ascending order.

Step 2: Label the k^{th} data point in the sorted order by $\lfloor \frac{k-1}{b} \rfloor + 1$.

Step 3: Put all the points that have the same label in one bin. If the last bin has size less than b, merge it with the second last bin (if exits). Let $\{B_1, B_2, \cdots, B_T\}$ be the set of bins obtained.

Step 4: For all points k in some bin B_i, define:

$$\hat{p}_i = \frac{1}{|B_i|} \sum_{i \in B_i} y_i$$

and

$$\hat{q}_i = \frac{1}{|B_i|} \sum_{i \in B_i} q_i$$

Step 5: The calibration score is calculated as:

$$CalibScore = \sqrt{\frac{1}{N} \sum_{i=1}^{T} |B_i| (\hat{q}_i - \hat{p}_i)^2}$$

3.2.2 Confidence interval estimation

The 95% confidence interval for $\hat{p}_i$ is approximated by:

$$\hat{p}_i \pm 1.96 \hat{se}_i$$

where $\hat{se}_i = \sqrt{\frac{\hat{p}_i(1-\hat{p}_i)}{|B_i|}}$ is the standard error at for the i^{th} bin.

It should be clear that the above formula is only an estimate of the confidence interval and thus does not guarantee true coverage. However, we found that this method is simple to implement and works well in practice.

In order to obtain the confidence interval for the calibration score, we use the following sampling procedure:

Data: A set of approximations $\{\hat{p}_1, \hat{p}_2, \cdots, \hat{p}_T\}$.

Parameter: Number of samples N_s.

Step 1: Sample N_s times. Each time, for i from 1 to T, draw $\hat{p}_i^* \sim \mathcal{N}\left(\hat{p}_i, \hat{se}_i^2\right)$.

Step 2: For each sample, calculate the current calibration score as:

$$\sqrt{\frac{1}{N}\sum_{i=1}^{T}|B_i|(\hat{q}_i - \hat{p}_i^*)^2}$$

Step 3: Report 95% confidence interval for the calibration score as:

$$CalibScore_{avg} \pm 1.96\hat{se}_{score}$$

where $CalibScore_{avg}$ and $\hat{se}_{score}$ are the mean and the standard error of the scores calculated from the samples.

3.3 Visualize calibration

To have a more general view of a model's degree of miscalibration, we can plot on the pairs $(\hat{p}_1, \hat{q}_1), (\hat{p}_2, \hat{q}_2), \cdots, (\hat{p}_T, \hat{q}_T)$ obtained from the adaptive binning procedure and visualize the *calibration curve* of the model. Calibration curve provides a fine-grained insight into the calibration behaviors in various prediction ranges. To be perfectly calibrated, the calibration curve has to coincide with the diagonal line "y = x", or the *perfect calibration curve* (PCC). At places where the curve lies above the PCC, the model predicts with less confidence then it should have done. Conversely, the model is overconfident where the curve is below the PCC.

An advantage of using the points obtained from the adaptive binning procedure in visualizing calibration is that the plot also captures the refinement aspect of the model. The distribution of points' x coordinates corresponds to the distribution of the model's predictions. On the other hand, if using equally spaced bins, one would need an extra plot to demonstrate that distribution.

3.4 Applications of calibration in NLP

Calibration analysis can easily be applied to analyze posterior predictions of models for structure prediction problems. In this setting, binary events are well-defined as polar queries on (sub)structures of the model such as single words, entity-spans or parsing subtrees. It is not necessary to test whether a model is calibrated for all types of queries. Depending on different downstream tasks, we want to have good calibration on different types of queries. Take syntactic parsing as an example. If the downstream tasks is a sentiment analysis tasks, it is important for the model to be reliable in predicting if a phrase is an adjective phrase. For a different application such as coreference resolution, we would care more about noun phrase's boundaries.

3.4.1 Sequence models

Logistic regression has been shown to give better calibrated probabilities than Naive Bayes (Niculescu-Mizil and Caruana, 2005), which makes us hypothesize that discriminative models would be better than generative models in posterior predictions. We take one step towards proving this hypothesis by examining a two popular classes of sequence models for POS tagging, Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs). Elegant dynamic programming algorithms are available for compute the marginal probability of different tag configurations of substrings of a sentence (Finkel et al., 2006). The forward-backward algorithm, often used in training

CRFs, provides an efficient way to compute the marginal probability for substrings of length 1 or 2.

We conduct the first experiment on a regular WSJ data set extracted from CoNLL-2011 (Pradhan et al., 2011). Predicting POS tagging on the WSJ data is a nearly solved problem. Choosing such an easy task, we would like to analyze the behavior of well-calibrated models. Next, we analyze the two models on a more challenging task, predicting POS for tweets (Gimpel et al., 2011). The difficulty of this task does not only come from the linguistic structure of tweets but also from the relatively small size of the data set. The queries we choose to test calibration on are in the form: "Is the current word an X?". In the first experiment, X is the "NN" tag. In the second experiment, we change to predicting the "V" tag since we find that predicting nouns in tweets is still easy for both models.

Importantly, since our focus is on model structure, we only use basic features for CRF in the comparison. Specifically, the feature set consists of only transition features (tag-tag) and emission features (tag-current word). By doing so, we disentangle the advantage of CRF's model assumptions from the advantage of it having more parameters than HMM. Later, we insert more advanced features to investigate effects of features on calibration.

3.4.2 Coreference resolution

Coreference resolution is the task of determining what entities are referred to by which linguistic expressions (mentions). It remains a challenging task in NLP with state-of-the-art techniques are around 60-70% accuracy Durrett and Klein (2013); Yang et al. (2015). In fact, it is still a debate on which coreference system is the state-of-the-art since the methods were tested on various data sets and the results were usually reported in two or three metrics. A negative impact of this fact is that each time a user wants to start up new application domain, he or she has to re-evaluate a lot of methods. Calibration analysis is suitable for coreference resolution for two reasons. First, when the accuracy scores are all low, a few percentages difference between imperfect models in accuracy are not significant since the models will make unsatisfiable mistakes eventually. At that time, it is more important to choose a model that is capable of truly quantifying its own mistakes so that risks can be estimated and possibly mitigated. Second, calibration analysis is flexible. The calibration query can be altered for different downstream goals as long as the tested model provides a mechanism for calculating confidence scores of the query.

We conduct a calibration analysis on the Berkeley coreference system (Durrett and Klein, 2013) on the CoNLL-2011 data set. The core of this system is a mention-ranking log-linear model that, for each mention, computes the confidence score of that mention referring to itself (singleton) and each of the previous mentions. Entity clusters are implicit during inference and only constructed after the relationships between the mentions have been determined. Unfortunately, due to this structure, the marginal probabilities of crucial queries such as "Does this pair of mentions belong to the same cluster?" is complicated to derive exactly. However, we can still obtain a fairly good estimate of the confidence scores via Monte Carlo sampling. Concretely, we sample directly the distribution ouput by the model and construct the set of clusters for each sample. The approximated confidence score for a query is the empirical proportion of the number of times the answer for the query is yes over a lot of samples.

CHAPTER 4

REPORT AND DISCUSSION OF RESEARCH RESULTS

4.1 Calibration hyper-parameters

Picking the optimal bin size can be done via cross-validation. However, since we have a fairly large amount of data points, we simply set the bin size to be 1000 for POS tagging, 400 for Tweet POS tagging and 10000 for coreference resolution.

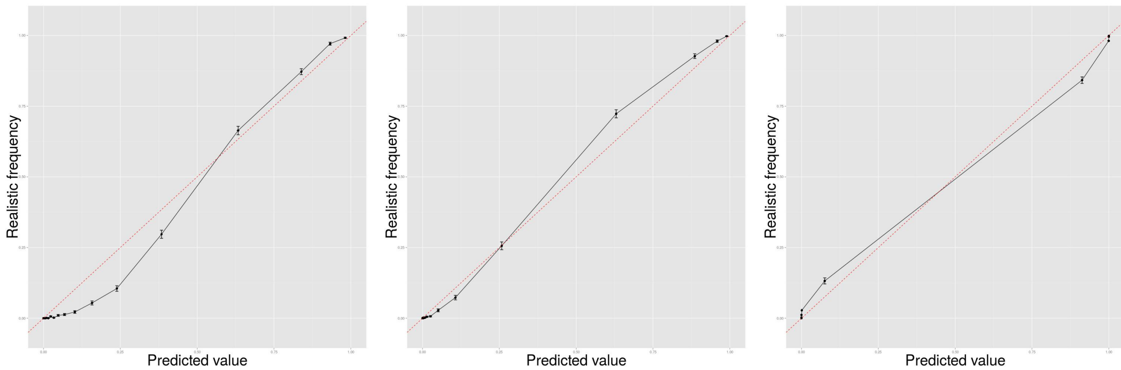
We report the calibration score and its confidence intervals for all our experiments. To compute the score, we implement the method described in section 3.2.2 with a sample of size of 1000.

4.2 Part-of-speech tagging

4.2.1 Data

We extract WSJ articles from the CoNLL-2011 dataset for this experiment. The original data set has already been split the into training, development and testing sets so we filter WSJ articles from each set and join the sentences into together. This process results in 11772 sentences for training, 1632 sentences for development and 1382 sentences for testing. Running the models on the testing set produces 30543 prediction-observation pairs. We then conduct calibration analysis on this set of pairs. The query tested is whether a word has the “NN” tag.

We train a HMM model using maximum likelihood principle. For CRF training, we implement the L2-regularized mini-batch AdaGrad method (Duchi et al., 2011). The batch size is chosen via cross-validation.



(a) (b) (c)

Figure 4.1: Calibration curves for HMM and CRF models (POS tagging) (a) HMM (b) CRF with basic features (c) CRF with advanced features

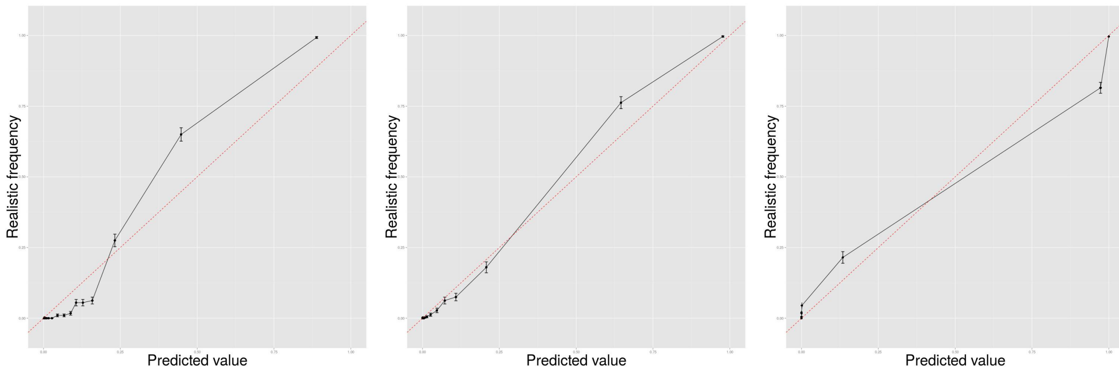
Model	Accuracy (%)	CalibScore
HMM	88.10	$0.042 \pm 5.27\text{e-}5$
CRF-Basic	93.02	$0.021 \pm 6.81\text{e-}5$
CRF-Advanced	96.08	$0.018 \pm 6.11\text{e-}5$

Figure 4.2: Accuracy and calibration scores of the models for POS.

4.2.2 Results

Firstly, we compare HMM with a CRF model with basic features (CRF-Basic). As mentioned in section 3.4.1, CRF-Basic contains only the transition features and the emission features. Figures 4.1(a) and 4.1(b) show calibration curves of the two models. CRF-Basic attains a significantly lower calibration score than HMM does (Figure 4.2). Moreover, the CRF-Basic's calibration has less mid-range points that of HMM, which indicates that the model generates more refined predictions.

To measure the affect of features on calibration, we add advanced features such as surrounding words, word shape, word length, prefixes and suffixes to CRF-Basic. We discover that as we use better features, we obtain more refined and calibrated predictions. The better featurized CRF (CRF-Advanced), which achieves a 96% accuracy on the task, produces a calibration curve just slightly off the PCC (Figure 4.1(c)).



(a) (b) (c)

Figure 4.3: Calibration curves for HMM and CRF models (TweetPOS) (a) HMM (b) CRF with basic features. (c) CRF with advanced features

Model	Accuracy (%)	CalibScore
HMM	67.55	0.075 ± 1.40e-4
CRF (Basic)	78.12	0.034 ± 1.61e-4
CRF (Advanced)	86.19	0.048 ± 1.47e-4

Figure 4.4: Accuracy and calibration scores of the models for TweetPOS.

4.3 Twitter part-of-speech tagging

4.3.1 Data

We repeat the same analysis in section 4.2 predicting POS tags for tweets. For this experiment, we use the ARK's Twitter POS data set (Gimpel et al., 2011), which consists of 1000 sentences for training, 327 sentences for development, 500 sentences for testing. The number of prediction-observation pairs obtain from testing set is 6160. Unlike the regular POS experiment, The query tested is whether a word has the "V" tag since we find that predicting "N" on tweets is still easy for the models.

4.3.2 Results

Comparing between HMM and CRF-Basic, we observe the same pattern as in the section 4.2. CRF-Basic gives a more calibrated and sharper predictions than HMM (Figure 4.3). In both experiments, the calibration score of HMM is always twice as large as that of CRF-Basic. On the other hand, unlike the previous experiment,

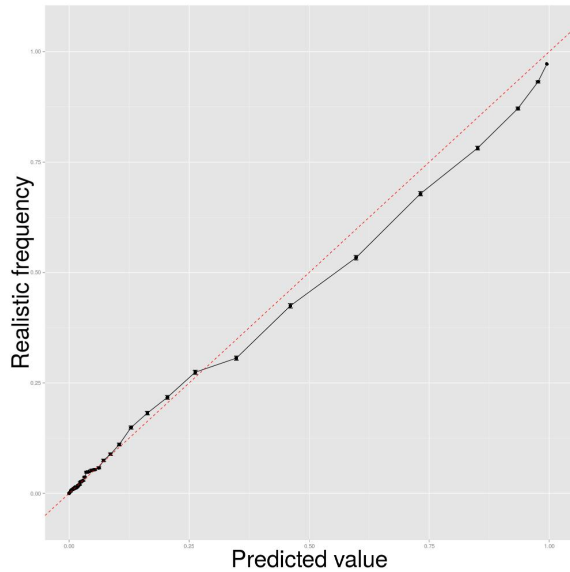


Figure 4.5: Calibration curve of Berkeley-coref.

adding more features to CRF worsens the calibration score (Figure 4.4). While CRF-Advanced produces sharper predictions, it is overconfident at high-score predictions. There are two possible explanations for this phenomenon. First, the data size is smaller; hence, the score may be not approximated well enough. Second, the feature set increases the recall but not the precision of the model.

4.4 Coreference resolution

4.4.1 Data

For this experiment, we use the available implementation of the Bekerley coreference resolution (Berkeley-coref) (Durrett and Klein, 2013). The data set used is the CoNLL-2011 data set. The performance score of this system was around 62% on the development set. The query tested is whether two mentions belong to the same entity. We apply calibration analysis on about 4 millions predictions generated from the development set.

4.4.2 Results

Although having a far-from-perfect accuracy score, the Berkeley's predictions are extremely reliable (Figure 4.5). Its calibration score is $0.007 \pm 6.32\text{e-}6$, much smaller than the models for the easier POS tagging task in the previous sections. The results also help explaining the effect of features on calibration. Berkeley-coref employs a carefully selected set of features called SURFACE. Whereas in the TweetPOS experiment, we see that adding features can degrade calibration, in this experiment, we observe a model with a good set of features can be very reliable despite making a lot of mistakes. The result of this experiment also confirms that discriminative models, with quality features, produces highly reliable predictions.

CHAPTER 5

CONCLUSIONS AND IMPLICATION FOR FUTURE RESEARCH

In this thesis, we have presented calibration analysis as a new framework for evaluating NLP probabilistic models. This framework directly measures the quality of posterior predictions rather than the quality of the final predictions. It is not only useful for investigating cascaded errors and estimating risks in NLP pipelines, whether the downstream tasks are stated explicitly or not. We propose two mechanisms to communicate the analysis results to the users and demonstrate procedures to construct them. Since calibration analysis is not a common practice in NLP research, we choose to introduce fairly simple procedure so that researchers can implement for themselves. However, for this reason, the approximation of miscalibration still has errors. Hence, the confidence intervals constructed are imperfect. In the future, we would like to reduce this errors by using more advance non-parametric regression methods such as local likelihood, whose theoretical bounds has been well-studied. On the other hand, we would like apply our calibration analysis to obtain more insights into the behaviors of probabilistic models. In our experiments, discriminative models tend give better calibrated predictions than generative models. But does this fact also affect by the nature of data? Besides that, we would like to know if calibration of a type of query can infer calibration of other related types of query. A another important question to ask was how calibration of a model would improve the performance of the entire pipeline. Answering this question requires setting up an end-to-end NLP pipeline and a study on recalibration methods.

CHAPTER 6

ACKNOWLEDGMENT

I want to dedicate this thesis to my loved ones.

This work can not be completed without the guidance of Prof. Brendan O'Connor. Natural Language Processing is a new field for me but his faith on my ability makes everything easier. The learning opportunities that he offered me were invaluable. Thanks to him, I was able to meet other great researchers in the field, who left deep impressions on me. I admire him for always patiently listen to me and giving the best advice. I am lucky to have him as my advisor.

I want to pay my respect to Prof. Erik Learned-Miller, who always takes good care of me during my four years of college. From the beginning, he appreciated my programming skills and guided me to my first research experience. He was a source of help whenever I ran into trouble. I will not forget your lessons about research and life, Professor!

I greatly appreciate the help of Caitlin Cellier. Discussing with her about errors in coreference resolution systems helped shaping my initial ideas for this work.

I want to thank my awesome friends. I am fortunate to befriend with too many bright people. Their successful stories teach me that success comes from anywhere as long as you appreciate your opportunities. I wish them all the bests in their future careers.

My family is a part of me. My parents always keep their eyes on my study progress. A few days ago, I still receive calls from them worrying about me not finishing this thesis. How many people on earth would receive that kind of luxury?

I am thankful that I have awesome parents. I hope this work would bring them happiness.

My sweetest words will be for my love, Huyen. When I am awake, she is asleep. When she sees the sun, I am already in my bed. But we still find time to encourage each other, and to practice generosity, tolerance and optimism together. We share the dream of being better people. The thesis is my humble attempt to fulfill that dream with her.

Finally, thanks UMass for being a peaceful home for my four years of college. I didn't like the school at first but now I love it! I am proud to be one of its students.

BIBLIOGRAPHY

Fabio Canova. Statistical inference in calibrated models. *Journal of Applied Econometrics*, 9(S1):S123–S144, 1994.

Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168. ACM, 2006.

Thomas F Cooley. Calibrated models. *Oxford Review of Economic Policy*, 13(3): 55–69, 1997.

Richard A Davis, Adrian J Charlton, John Godward, Stephen A Jones, Mark Harrison, and Julie C Wilson. Adaptive binning: An improved binning method for metabolomics data using the undecimated wavelet transform. *Chemometrics and intelligent laboratory systems*, 85(1):144–154, 2007.

Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *The statistician*, pages 12–22, 1983.

David Draper. Assessment and propagation of model uncertainty. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 45–97, 1995.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

Greg Durrett and Dan Klein. Easy victories and uphill battles in coreference resolution. In *EMNLP*, pages 1971–1982, 2013.

Greg Durrett and Dan Klein. A joint model for entity analysis: Coreference, typing, and linking. In *Proceedings of the Transactions of the Association for Computational Linguistics*, 2014.

Jenny Rose Finkel, Christopher D Manning, and Andrew Y Ng. Solving the problem of cascading errors: Approximate bayesian inference for linguistic annotation pipelines. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 618–626. Association for Computational Linguistics, 2006.

Markus Frölich. Non-parametric regression for binary dependent variables. *The Econometrics Journal*, 9(3):511–540, 2006.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 42–47. Association for Computational Linguistics, 2011.

Liang Huang and David Chiang. Better k-best parsing. In *Proceedings of the Ninth International Workshop on Parsing Technology*, pages 53–64. Association for Computational Linguistics, 2005.

Wee Kheng Leow and Rui Li. The analysis and applications of adaptive-binning color histograms. *Computer Vision and Image Understanding*, 94(1):67–91, 2004.

Robert G Miller. *Statistical prediction by discriminant analysis*, volume 4. American Meteorological Society, 1962.

Allan H Murphy. A new vector partition of the probability score. *Journal of Applied Meteorology*, 12(4):595–600, 1973.

Allan H Murphy. What is a good forecast? an essay on the nature of goodness in weather forecasting. *Weather and forecasting*, 8(2):281–293, 1993.

Allan H Murphy and Robert L Winkler. Probability forecasting in meteorology. *Journal of the American Statistical Association*, 79(387):489–500, 1984.

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 625–632. ACM, 2005.

Naomi Oreskes, Kristin Shrader-Frechette, Kenneth Belitz, et al. Verification, validation, and confirmation of numerical models in the earth sciences. *Science*, 263 (5147):641–646, 1994.

Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–27. Association for Computational Linguistics, 2011.

Donald B Rubin. Bayesianly justifiable and relevant frequency calculations for the applies statistician. *The Annals of Statistics*, 12(4):1151–1172, 1984.

Sameer Singh, Sebastian Riedel, Brian Martin, Jiaping Zheng, and Andrew McCallum. Joint inference of entities, relations, and coreference. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 1–6. ACM, 2013.

Charles Sutton and Andrew McCallum. Joint parsing and semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages 225–228. Association for Computational Linguistics, 2005.

Kristina Toutanova, Aria Haghighi, and Christopher D Manning. Joint learning improves semantic role labeling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 589–596. Association for Computational Linguistics, 2005.

Larry Wasserman. *All of nonparametric statistics*. Springer Science & Business Media, 2006.

Ben Wellner, Andrew McCallum, Fuchun Peng, and Michael Hay. An integrated, conditional model of information extraction and coreference with application to citation matching. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 593–601. AUAI Press, 2004.

Bishan Yang, Claire Cardie, and Peter Frazier. A hierarchical distance-dependent bayesian model for event coreference resolution. *arXiv preprint arXiv:1504.05929*, 2015.