

CALIBRATION 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 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**

Statistical natural language processing (NLP) models assign a posterior distribution to the set of possible outcomes and make their predictions based on those scores. Current performance metrics for NLP systems only take into account the model final decisions, which not only depends on the quality of the systems model but also the inference scheme, rather than explicitly reflect the quality of the prediction posterior distribution. In this thesis, I propose a metric for directly examining the quality of the posterior distribution, the calibration test. First of all, I will present the theoretical foundation for the concept of calibration. After that, I apply the test to numerous families of NLP models and show that the calibration is complementary to the traditional metrics in the sense that it provides a more comprehensive insights into the performances of NLP systems.

CHAPTER 1

INTRODUCTION

The ambiguity of natural language motivated the use of the probabilistic model for tackling natural language processing (NLP) tasks. 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 F1 score also solely measure how 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 does 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 scores 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 reliability 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. In Rubin (1982), it was argued that the 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 biochemistry 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 $S = \{(q_1, y_1), (q_2, y_2), \dots, (q_n, y_n)\}$ 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:

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

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:

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

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 (1982), 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:

$$f(x) = pg_1(x) + (1 - p)g_2(x) \tag{3.2}$$

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

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

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 this quantity by the *MSE calibration score* or simply calibration score, interchangeably.

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 MSE 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, non-parametric 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* (CITE 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 $\{(q_1, y_1), (q_2, y_2), \dots, (q_n, y_n)\}$.

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 exists). Let $\{B_1, B_2, \dots, B_T\}$ 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 MSE calibration score is calculated as:

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

3.2.2 Confidence interval estimation

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

$$\hat{p}_i \pm 1.96 \hat{s}e_i$$

where $\hat{s}e_i = \sqrt{\frac{\hat{p}_i(1-\hat{p}_i)}{|B_i|}}$ 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 MSE score, we use the following sampling procedure:

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

Parameter: Number of samples N_s .

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

Step 2: Report 95% confidence interval for the calibration score as:

$$CalibMSE_{avg} \pm 1.96\hat{se}_{MSE}$$

where $CalibMSE_{avg}$ and \hat{se}_{MSE} are the mean and the standard error of the MSE calibration 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 $(\hat{p}_1, \hat{q}_1), (\hat{p}_2, \hat{q}_2), \dots, (\hat{p}_T, \hat{q}_T)$ obtained from the adaptive binning procedure and visualize the *calibration curve* of the model (Figure BLAH BLAH). Calibration curve provides a fine-grained insight into the behavior of the model. To be perfectly calibrated, the curve has to coincide with the diagonal line “ $y = x$ ”, or the *perfect calibration curve* (PCC). At places where the calibration curve lies above the PCC, the model predicts with less confidence then it should have done. Conversely, the model is overconfident where the calibration curve lies 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 is model-indepent and 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 subtree. It is not necessary to test whether a model is calibrated for all types of queries. Depending on different downstream taks, we want to have good calibration on different types of queries. Take syntatic 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 (CITE Caruana), which is a signal that discriminative models would be better than generative models in posterior predictions. We take one step towards proving this claim by examining a two popular classes of sequence models for POS tagging, Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs). First of all, we conduct an experiment on a regular WSJ data set extracted from CoNLL-2011 (CITE ConLL). Predicting POS tagging on the WSJ data set is almost solved problem (CITE somebody). Choosing such an easy task, we would like to analyze the behavior of well-calibrated models. Next, we test the two models on a more challenging task, predicting POS for tweets (CITE Noah). 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 a(n) 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. Elegant dynamic programming algorithms are available for compute the marginal probability of a particular tag configuration of a substring in the sentence (CITE Finkel?). The forward-backward algorithm, used in training CRF, provides an efficient way to compute the marginal probability for substrings of length 1 or 2.

Also, since our focus is on model structure, we only use basic features for in our CRF implementation. Our 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 also conduct an ablation test on the CRF to verify the affect of features on calibration.

3.4.2 Coreference resolution

Coreference resolution is the task of BLAH BLAH. It remains a challenging task in NLP with state-of-the-art techniques are around 60-70% accuracy on the CoNLL data set. In fact, the performances of those techniques, even on the same data set, are often not comparable due to the lack of an evaluating procedure for coreference resolution. Selecting one techinque for a end-to-end system is a really a matter of tradeoff and depends on the nature of the downstream tasks. 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. This is when calibration analysis comes to help as a tool to separate the reliable imperfect models from the non-reliable ones. 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 the confidence level of the query.

We conduct a calibration analysis on the Berkeley coreference system (CITE Klein) 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 output 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 MSE calibration score and its confidence intervals for all our experiments. To compute the score, we implement the sampling 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 splitted 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 choosen by cross-validation.

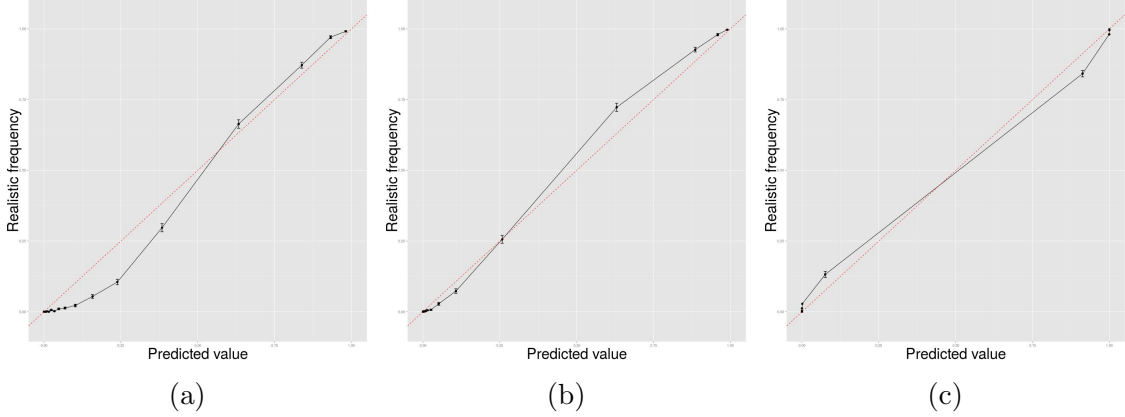


Figure 4.1: Calibration curves for HMM and CRF models (POS tagging) (a) HMM (b) CRF with basic features (c) CRF with advanced features

Model	Accuracy (%)	CalibMSE
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. Figure 4.1 shows calibration curves of the two models. CRF-Basic attains a significantly lower MSE calibration score than HMM does (0.019 vs. 0.035). Moreover, it also produces more refined predictions than those of HMM, as seen from the distributions of points along the x-axis.

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 MSE calibration score than HMM does (Figure 4.4). Moreover, the CRF-Basic’s calibration has less mid-range points than of HMM, which indicates that the model generates more refined predictions.

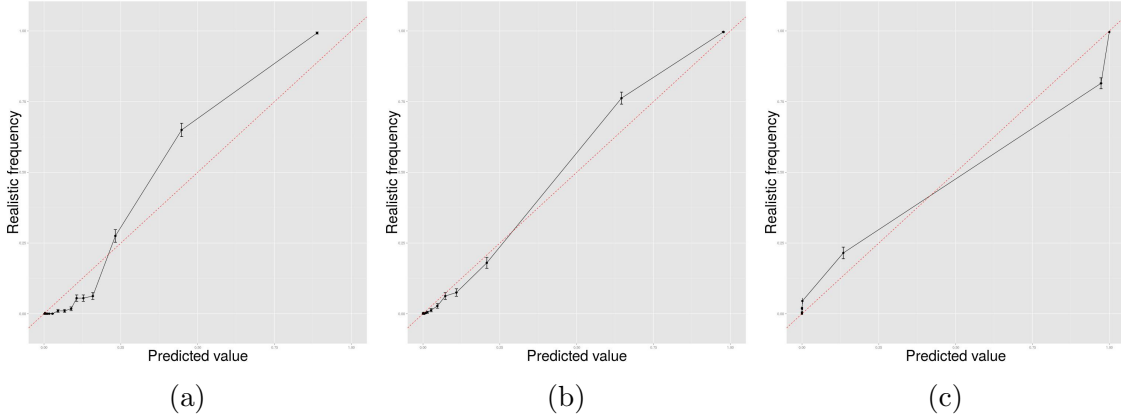


Figure 4.3: Calibration curves for HMM and CRF models (TweetPOS) (a) HMM (b) CRF with basic features. (c) CRF with advanced features

Model	Accuracy (%)	CalibMSE
HMM	67.55	$0.075 \pm 1.40\text{e-}4$
CRF (Basic)	78.12	$0.034 \pm 1.61\text{e-}4$
CRF (Advanced)	86.19	$0.048 \pm 1.47\text{e-}4$

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

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% accuraccy on the task, produces a calibration curve just slightly off the PCC.

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 4.4). In both experiments, the MSE calibration score of HMM is always twice as large as that of CRF-Basic. On the other hand, unlike the previous experiment, 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.

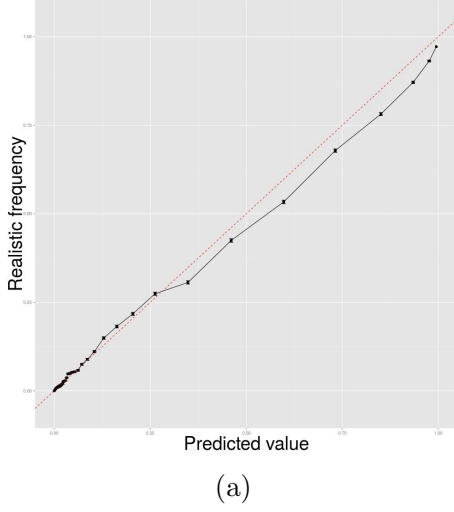
4.4 Coreference resolution

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 system’s predictions are extremely reliable (Figure). Its calibration score is even much smaller than the models in the POS task, which is a substantially easier task. The results also help explaining the effect of features on calibration. Berkeley-coref employs a carefully selected set



Model	CalibMSE
Berkeley-coref	$0.007 \pm 6.32\text{e-}6$

(b)

Figure 4.5: Calibration analysis results for Berkeley-coref (a) Calibration curve (b) Calibration score

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. Last but not least, throughout three experiments, it is not difficult to recognize that discriminative models, with a quality features, tend to be very well-calibrated.

CHAPTER 5

**CONCLUSIONS AND IMPLICATION FOR FUTURE
RESEARCH**

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.
- 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.

- 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 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.