

# Comprehensive Exam

## Topic: Theoretical Models for Natural Language Processing

Mihai Rotaru  
Computer Science Department  
University of Pittsburgh

### 1. Introduction

Processing and generating natural language requires an understanding of how complex knowledge sources interact (morphologic, lexical, syntactic, semantic, discourse, pragmatic etc.). At a first look, natural language might seem rather regular thus warranting a rule-based approach. Rooted in linguistic theories of language, early work has used rule-based systems. In general, the rules were devised by hand thus creating and maintaining such a system was a tedious task. But, at a closer look, natural language abounds in exceptions and complex phenomena that are hard to model using rules. In the past 15 years, statistical approaches to Natural Language Processing (**NLP**) have gain momentum and proved successful for a variety of tasks. Their main advantage lies in the fact that probabilistic models handle well both rules and exceptions to those rules. Moreover, the models can be learned from data with minimum human intervention (though a time consuming annotation process is usually required).

But statistical approaches for NLP have a big drawback: a large quantity of (annotated) data is needed. Given the high cost of human annotation, this results in a data sparsity problem. Typically, in statistical NLP either a joint or a conditional distribution is estimated from data and used by the model. There are various techniques to circumvent the data sparsity problem. Previous work has extensively employed techniques like approximating the model with simpler models (e.g. using chain rule and n-grams for language modeling), incorporate prior knowledge (the Bayesian approach), etc.

Here, we look at Maximum Entropy (**ME**) as a technique to effortlessly incorporate multiple dependent or independent knowledge sources. We also look at techniques that take advantage of the inherent dependencies and independencies between the variable being modeled: Bayesian Belief Networks (**BBN**) and their time-aware simplifications Hidden Markov Models (**HMM**). We discuss how these models were applied to a variety of NLP tasks.

### 2. Maximum Entropy (ME)

The ME approach permits the construction of a probability distribution that incorporates multiple knowledge sources. Typically, the knowledge sources are represented as constraints that the target probability distribution has to satisfy. Since in most cases there are a large number of distributions that satisfy those constrains, the ME principle states that we should pick the distribution(s) that *assume nothing else*. For example, if there are no constraints on our distribution, “assume nothing else” principle requires that we should assume all outcomes as equally likely. Thus according to the ME principle, when no constraints are placed on our distribution, we should always choose the uniform distribution (i.e. the flattest distribution). But as the number of constraints increases, analytically computing the ME

distribution(s) becomes more complicated (see Section 2 of (Berger, Pietra, & Pietra, 1996)). Entropy measures how close a probability distribution is to the uniform distribution: the higher the entropy, the closer is our distribution to the uniform distribution. Thus, the “assume nothing else” principle can be recast as finding the distribution(s) with the maximum entropy (therefore the “Maximum Entropy” name).

Encoding knowledge sources as constraints is done using non-negative *feature functions* (also known as feature or selector functions (Rosenfeld, 1994)). Next, the expected value of each feature function under the target distribution is required to be equal to a certain value which is typically computed as the expected value of that feature function under the empirical distribution (computed from the training data). Given consistent constraints, a *unique* ME distribution is guaranteed to exist and has the following log-linear form:

$$P(x) = \frac{1}{Z_\lambda} \exp\left(\sum_i \lambda_i f_i(x)\right)$$

where  $f_i$  are the features functions,  $\lambda_i$  are weights we need to estimate and  $Z_\lambda$  is a normalizing constant. The  $\lambda_i$  parameters can be estimated using the Generalized Iterative Scaling algorithm (**GIS**) (Darroch & Ratcliff, 1972) or its more efficient version, the Improved Iterative Scaling algorithm (Berger et al., 1996). Parameters  $\lambda_i$  can be interpreted as follows: if  $\lambda_i > 0$  then for all  $x$  for which the feature is non-zero, the knowledge source increases the probability of seeing  $x$ . If  $\lambda_i < 0$  then for all  $x$  for which the feature is non-zero, the knowledge source decreases the probability of seeing  $x$ . If  $\lambda_i = 0$  then the knowledge source does not contribute to estimating the probability of seeing  $x$ . This suggests ME as a feature selection technique to understand the role a various knowledge sources for the NLP task at hand.

As described above, ME estimates joint distributions. Joint distributions are used in tasks like language modeling with applications in speech recognition and machine translation. (Rosenfeld, 1997) shows how n-gram information (unigram, bigrams and trigrams in his case) can be combined using ME for language modeling. But the ME framework can also be extended for conditional distributions. For most NLP tasks we are interested in estimating conditionals: given some context/history information we want to know the distribution over a target variable. Conditional ME approaches to various NLP tasks yielded performance figures similar to or better than state-of-art for several NLP tasks: part-of-speech tagging (**POS**) (Ratnaparkhi, 1996), word sense disambiguation (**WSD**) (Chao & Dyer, 2002), machine translation (**MT**) (Berger et al., 1996; Och & Ney, 2002; Sato & Nakanishi, 1998).

Several ME advantages make this technique appealing for NLP applications: it can combine multiple knowledge sources, the knowledge sources need not be independent, a ME model can be easily extended with additional knowledge sources, implicit smoothing. But a number of computation bottlenecks have limited the widespread application of ME: computing the normalizing constant, computing the expected value of feature functions at each step in the GIS algorithm, selecting the right feature functions, overfitting. We will discuss in details each pros and cons of ME in the following paragraphs.

Any number of knowledge sources can be incorporated using the ME framework. All we need to do is create the right feature functions for those knowledge sources and specify for each feature function the constraint on the expected value of that feature under the estimated distribution. For example, (Rosenfeld, 1997) uses information about unigram, bigram and trigram counts for language modeling. He also acknowledges that additional knowledge sources can be used for better language modeling: sentence length, syntactic information, dialogue variables (in the case of (spoken) dialogue systems). Specific properties of a word have been combined using ME for various word-level NLP tasks: the prefix, suffix and capitalization were used by (Ratnaparkhi, 1996) for POS; the word POS, POS class and suffix were used by (Chao & Dyer, 2002) for WSD; token prefix, suffix, capitalization and whether the token was a abbreviation were used by (Reynar & Ratnaparkhi, 1997) for sentence boundary detection; the POS of the source and target word were used by (Sato & Nakanishi, 1998) for estimating the word translation model.

Local knowledge sources were often augmented with various contextual knowledge sources: the word window (Berger et al., 1996; Chao & Dyer, 2002; Pakhomov, 2002; Ratnaparkhi, 1996; Reynar & Ratnaparkhi, 1997; Sato & Nakanishi, 1998), document information (Pakhomov, 2002), structurally related words via dependency trees (Chao & Dyer, 2002). (Berger et al., 1996) show how existing models that do not take into account context can be augmented with ME models that use context for tasks like word translation modeling, segmentation and word reordering. Even very complex information sources can be effortlessly combined using ME. For machine translation, (Och & Ney, 2002) show how the ME framework can be used to combine a source channel based model with additional knowledge sources like conventional lexicons, sentence length models and refined target language models.

Modeling dependent knowledge sources has always been a big challenge for statistical NLP. Given the large number of dependent factors that need to be accounted for, modeling a full joint is impractical due to data sparsity and computational considerations. To circumvent these problems, NLP models are usually broken down in sub-models (e.g. breaking down language modeling using the chain rule). To better estimate these sub-models several independence assumptions are being made that reduce the number of dependencies (e.g. the Markov assumptions for n-grams). In most cases these assumptions are clearly flawed (e.g. a word in a sentence is not always dependent only on the previous 2-3 words). Moreover, they can lead to suboptimal models: (Och & Ney, 2002) argues that the source channel approach to statistical machine translation is optimal only if the language and the translation models are equal to the true models. In contrast, the ME framework has no requirement that the knowledge sources should be independent. Many of the ME applications discussed above successfully employ dependent knowledge sources: the words in a word window, the prefix and suffix of a word, various n-grams etc.

Another big challenge for NLP applications is their scalability: how easy will the application make use of additional knowledge sources that becomes available after the model was designed. Typically, most NLP models are very rigid to incorporating additional knowledge. In contrast, in ME modeling, additional knowledge can be easily incorporated: all we have to do is to design the new feature functions, add them to the ME model constraints and learn a new set a weights for the new model. The ease of adding additional knowledge sources in the ME framework has allowed researchers to evaluate the contribution of each source or the contribution of the combination of specific sources. (Rosenfeld, 1997) shows that incrementally adding higher level n-grams results in better quality sentences in a ME language modeling. (Chao & Dyer, 2002) show that adding more contextual information (structurally related words) helps for WSD. (Och & Ney, 2002) shows that adding to a standard statistical MT system knowledge of sentence length, conventional lexicons and better target language models increases the translation quality with respect to a plethora of metrics. Interestingly, they also show that the source channel model for machine translation is a particular case of ME: a ME model with two features (the language model and translation model). In addition, the ME interpretation of the source channel model also optimizes the model scaling factors an optimization process that is skipped in the source channel approach.

Whenever probabilistic models are used in NLP, data sparsity is a big problem. Complex smoothing techniques are used to increase the quality of the empirical distribution derived from data (see (Chen & Goodman, 1996) for a comparison of smoothing methods for n-gram language modeling). For ME models, smoothing of the estimated probability distribution is performed implicitly. The fact that we are using the probability distribution with the highest entropy is responsible for this implicit smoothing. Please note that this is different from smoothing the actual ME constraints we derive from the training data (i.e. the expected value of feature functions under the empirical distribution). For details on smoothing techniques for ME constraints see (Chen & Rosenfeld, 1999).

While the ME approach has a number of properties that make it very appealing for NLP tasks, it also has several disadvantages. In general, using joint distributions over many variables poses problems in terms of the representation space as well as the high computational cost of making inferences. In the particular case of ME framework, there are additional computational bottlenecks. In the learning phase, for each iteration, we need to compute for each feature the expected values under the current distribution. This

process can be time consuming and sometimes intractable (e.g. sentence level language modeling (Rosenfeld, 1997)). (Rosenfeld, 1997) circumvent this problem by estimating the expected value using Gibbs sampling. While less time consuming, this approach limits the optimality of the learned ME model.

Computing the normalizing constant ( $Z_\lambda$ ) is also another computational bottleneck. Again, this can be avoided using sampling when training the model (Och & Ney, 2002). After the ME model is learned, for certain types of problems, the normalization is not required because the model is used in an *argmax* expression. ME models for machine translation (Och & Ney, 2002) or for language modeling (Rosenfeld, 1994) are examples of cases where the normalization step can be omitted.

Since ME modeling can be viewed as maximum likelihood training for log-linear models, ME modeling is prone to overfitting. This can happen especially when a large number of features is used. Previous work in ME modeling abounds in feature templates: authors devise a series of templates that might capture useful information and the training set is used to instantiate this templates and create feature functions (Berger et al., 1996; Chao & Dyer, 2002; Ratnaparkhi, 1996; Sato & Nakanishi, 1998). These templates result in a large number of features which can lead to overfitting. Cutoffs or the Basic Feature Selection algorithm (Berger et al., 1996) are used to select the most relevant features and prevent overfitting.

A number of implementations of ME are available: *YASMET* (<http://www.fjoch.com/YASMET.html>) developed by Franz Joseph Och and its extension that allows for feature selection *yasmetFS* (<http://www.isi.edu/~ravichan/YASMET/>) and the *OpenNLP MaxEnt* Java package available at (<http://maxent.sourceforge.net/>).

### 3. Bayesian Belief Networks (BBNs)

BBNs provide a compact and efficient representation of a joint distribution. They take advantage of the inherent independencies that exist between variables. A direct acyclic graph is used to encode the independence/dependence information and is usually referred as the network structure. For each node in the graph, a conditional probability table has to be specified; it encodes how the node's distribution is influenced by the parents' values. But there is a price we have to pay for the efficient representation: exact inference in BBN is exponential in number of nodes. All exact algorithms are instantiations of recursive factoring. For certain classes of BBN structures, efficient algorithms exist (e.g. Pearl's belief propagation algorithm for polytrees is linear in network parameters). Approximate inference algorithms are based on various sampling methods. A number of algorithms for learning the BBN structure and/or its parameters from data are also available.

Since many independencies/dependencies exist between concepts in NLP or are forced due to specific simplifying assumptions, BBNs are appealing for modeling NLP tasks. In BBN modeling, knowledge can be introduced in two ways: in the structure of the network and in the network parameters. Below, we will discuss how the structure and the parameters of a BBN were design for specific NLP applications.

Three approaches to designing the network structure were used in previous work: designing the structure by hand, using external resource to generate the structure or learning the structure from data. Whenever the network structure was designed by hand, researchers used their intuition and knowledge about variables being modeled or made certain simplifying assumptions. (Keizer, Akker, & Nijholt, 2002) encode in their network dependencies like the dependency of the current user dialogue act on the previous system and user act and independencies (through simplifying assumptions) like the independence of the lexical form of a user turn and the previous history given the current user dialogue act. A Naïve Bayes structure is used in (Wai, Meng, & Pieraccini, 2001) for dialogue modeling; the assumption is that the concepts present in a user turn are independent of each other given the user intention. The entire domain task was encoded in a BBN in the Bayesian Receptionist system by (Horvitz & Paek, 1999). In their system, the BBN representation of the task allowed the system to reason about the information (evidence) available so far and to predict the next system move (acquire more information or proceed to a specific

subtask). (Zukerman, McConachy, & Korb, 1998) design their BBNs to capture sound inferences in the NAG argumentation system. (Narayanan & Jurafsky, 1998) encode the independency between syntactic and thematic context assuming that their dependencies are explicitly captured by specific constructions (e.g. Main Verb or Reduced Relative).

In other studies, researchers have used external information sources to design the network structure. The WordNet hyponym hierarchy has been extensively used for tasks like unsupervised learning of verb selectional preferences (Ciaramita & Johnson, 2000), adjective sense disambiguation (Chao & Dyer, 2000) or question answering (Ramakrishnan, Jadhav, Joshi, Chakrabarti, & Bhattacharyya, 2003). In these studies, the hyponym hierarchy was used to propagate evidence regarding words from the training data to their ancestor concepts or other similar words. In (Galley, McKeown, Hirschberg, & Shriberg, 2004) the intuition that the agreement/disagreement label of the current speaker turn is dependent on the previous exchange with the speaker the current speaker is responding to, was captured by using the adjacency pairs structure learned beforehand. Given the large amount of data needed to learn the network structure, a limited number of applications of BBNs for NLP have tried learning the network structure and only for a small number of variables (Keizer et al., 2002).

In previous BBN-based NLP studies, network parameters were either designed by hand or learned from a corpus. In (Ciaramita & Johnson, 2000) the parameters were designed so that the network inference will result in a “explain away” effect of word classes (discussed in more detailed below). Mapping between frequency of occurrence in the data and probability intervals is used in (Wai et al., 2001). Noisy-OR nodes are used by (Ramakrishnan et al., 2003) for a question answering system. But in many cases, the parameters are estimated from the data. In general maximum likelihood or maximum a posteriori estimations are being used (Chao & Dyer, 2000; Keizer et al., 2002; Narayanan & Jurafsky, 1998). In (Galley et al., 2004) Maximum Entropy robustness to dependent knowledge sources and data sparsity is used to improve the parameter estimation. In general, parameters learned from data resulted in a better performance compared with hand coded parameters (Ramakrishnan et al., 2003; Wai et al., 2001).

While BBN modeling is suitable for many NLP applications, the high computation cost of BBN inference limits its application. To alleviate this problem, typically, researchers limit the number of variables being modeled or the density of the network structure. In (Chao & Dyer, 2000; Ciaramita & Johnson, 2000; Ramakrishnan et al., 2003) only the part of the WordNet hyponym hierarchy was used when modeling. (Zukerman et al., 1998) uses a focusing mechanism to limit the number of BBN variables (propositions in their case) taken into account when producing the Argument Graph. To limit the density of the network structure, independence assumptions are usually made even though these assumptions are not true in general. For example, (Wai et al., 2001) uses a Naïve Bayes model that assumes independence between concepts present in a turn given the user goal.

Besides reasoning about variables given evidence, BBN were also used in NLP tasks to satisfy other modeling needs. (Berger et al., 1996) illustrate how Maximum Entropy offers a way to enforce a simple paradigm under complex conditions (e.g. distribute the probability mass uniformly in absence of information). Similarly, BBN have been used to enforce simple paradigms in complex interactions. (Ciaramita & Johnson, 2000) use noisy-OR nodes in a BBN network to encode a simple paradigm: for every word if at least one of the words’ parent classes is part of the selectional restrictions of a verb then the word is likely to be part of those selectional restrictions too. The resulting BBN network that superimposes this simple paradigm on WordNet hyponym hierarchy exhibits an “explain away” effect: specific word parents’ classes are removed from the selectional restrictions automatically when learning from data.

In (Narayanan & Jurafsky, 1998) BBN were used to produce a computation model of the human performance on the “garden-path” effects (e.g. “The horse raced past the barn fell”). Their model offers insight into the cognitive process of human sentence processing. Since their model can predict accurately the recorded human behavior, the BBN structure properties can be used to infer properties of human

sentence processing. In their case, their BBN structure suggests that various knowledge sources (syntax, thematic roles) are processed independently by humans and that they are probabilistically combined at a latter stage to arrive to an interpretation of the sentence. The range of inferences that can be done in BBN network (setting, changing or removing evidence) allowed (Zukerman, McConachy, & Korb, 1999) to implement various exploratory argumentation operations (adding a proposition, “what if” construct, the effect of a proposition on the argument, etc.). In (Wai et al., 2001) the same BBN was used for two distinct tasks. One was to detect the user goal based on the concepts present in the user turn (bottom-up inference). The other was to detect missing or spurious concepts (top-down inference) which were used for guiding the next system move.

## 4. Hidden Markov Models (HMMs)

HMMs can be viewed as a special case of BBNs. But, while BBNs’ main focus is on modeling joint probabilities, HMMs are appropriate for modeling sequences (i.e. time-aware phenomena). More specifically, HMMs can be used to model sequences where there are interactions between members of the sequence (Rabiner, 1989). Since in many NLP tasks we have to model such sequences (e.g. the words that form a sentence, the sequence of POS tags in a sentence, the sentences in a discourse, etc.), HMMs are extensively used in the NLP literature.

An HMM is characterized by a set of states, a set of observations, a transition probability distribution, an observation probability distribution and an initial state distribution (Rabiner, 1989). Figure 1 shows an example of an HMM and a BBN view of a time slice. HMMs are a generative model. Each state emits an observation according to the observation probability distribution. After that, then the next state is selected according to the transition probability distribution and the process is repeated.

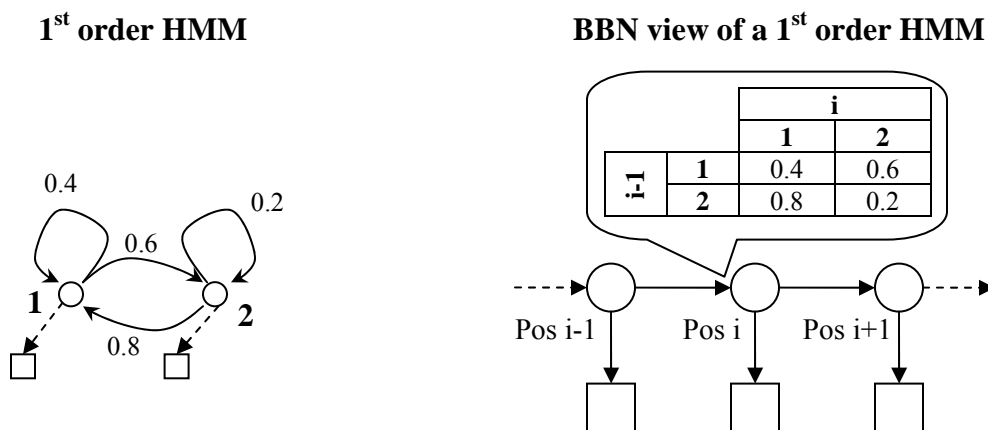


Figure 1. A 1<sup>st</sup> order HMM and its BBN view.

There are many extensions to HMMs as described before. These extensions were usually designed to accommodate specific requirements of the tasks being modeled. In the classic HMMs (1<sup>st</sup> order HMMs) the transition probability is dependent only on the current state (modeled as the horizontal dependency arches in the BBN view). This is also known as the Markov assumption. One way to extend HMMs is to enlarge the number of previous states the transition probability is dependent on. In an N<sup>th</sup> order HMM, the transition probability is dependent on the previous N states, thus more context is used in making the decision about the next state. Figure 2 shows the BBN view of a 2<sup>nd</sup> order HMM. Note that a graphical representation of the HMM network is not possible anymore. N-th order HMMs were used by previous work for a better approximation of the underlying process: POS tagging (Brants, 2000), word segmentation and classification (Law & Chan, 1996), text chunking (Skut & Brants, 1998). Other extensions address

some of the limitations of classic HMMs (Rabiner, 1989). Variable duration HMMs attempt to fix the geometric distribution of staying in the same state which is inappropriate for many applications (e.g. signal processing) (Rabiner, 1989). In null transition HMMs the empty observation can be emitted thus extending the labeling capabilities of HMMs for structure discovery (Bikel, Schwartz, & Weischedel, 1999).

### BBN view of 2<sup>nd</sup> order HMM

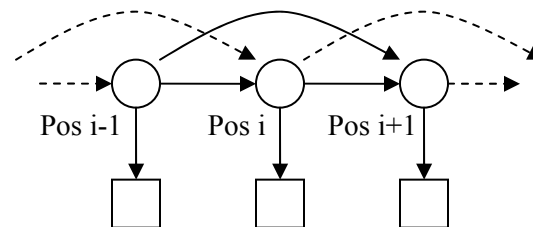


Figure 2. The BBN view of a 2<sup>nd</sup> order HMM.

Designing the right set of states and observations is a crucial first step for a successful application of HMMs. In the classic HMM approach, a small number of states and a larger set of observations is being used. In (Brants, 2000), their states represent POS tags while the observations are all words in the vocabulary; (Law & Chan, 1996) uses a small number of word classes (192) as HMM states and sequences of Chinese characters as the observations for a word segmentation task. Word positions in an English sentence are used as states and the parallel French sentence's words are used as observations in (Vogel, Ney, & Tillmann, 1996) to generate the best word alignment between a English and a French sentence. In these approaches, each state has a clear definition (e.g. a POS tag, a word class, etc). A completely different approach is taken in (Barzilay & Lee, 2004). They use topics as states and sentences as observations for content structure modeling task. But since they do not know the topics beforehand, an iterative process is used to change the state "meaning" (as defined by state's observation probability distribution) until the "right" topics are discovered.

This "few-states-many-observations" modeling approach is appropriate for cases where a label spans a single observation. But in many tasks, a single label spans more than one observation. For example, the text chunking (Skut & Brants, 1998) and name entity recognition (Bikel et al., 1999; Zhou & Su, 2002) tasks can be modeled so that a label will span over multiple words. For these cases, a large state space and a small number of observations approach is usually used. The states are represented as tuples and the observations are the values of a certain dimension from the tuples. Typically, a structural dimension is also used in the tuples; it captures where a label starts and where it ends. Interestingly, observations are emitted deterministically in this approach: the state emits deterministically the observation encoded in the tuple. Please note that these HMMs are not observable Markov processes because, in the former, there are states that emit the same observation which is not allowed in the latter. Intuitively, in this approach, the HMM structure is composed of a set of "sub-HMMs" (one for each label) with certain restrictions regarding transitions from a state in a sub-HMM to a state in another sub-HMM.

Estimating the right transition and observation probability distributions is also very important. Approaches range from manually designing the distributions to learning them from a tagged corpus. In (Vogel et al., 1996), the transition probability distribution was designed manually to encode an interesting observation regarding word alignments in statistical machine translation: word alignments tend to have a localization effect for specific pairs of languages. For the HMM modeling approach discussed in the previous paragraph, the observation probability distribution is forced by the model. But in the majority of cases, both the transition and the observation probability distributions were learned from an annotated corpus. Typically back-off smoothing techniques (Bikel et al., 1999; Zhou & Su, 2002) (especially for

larger order HMMs (Brants, 2000; Law & Chan, 1996; Skut & Brants, 1998)) and smoothed counts (Barzilay & Lee, 2004) are used to account for data sparsity. In (Barzilay & Lee, 2004) because their observations set is very large (all possible sentences), bigrams are used to encode the observation probability distribution for each state.

Handling unknown observations is also important for HMM approaches to NLP. For example, in POS tagging, a tagger will most certainly be faced with unknown words in the test set. To address the unknown words problem, (Brants, 2000) uses word suffixes based on the intuition that for inflected languages the suffix is a strong predictor of POS (e.g. the “able” suffix for adjectives). (Bikel et al., 1999) assigns all words to a new observation (UNKOWN-WORD) and explicitly learns how this observation interacts with known words for a name entity recognition and classification task.

Whenever a large state or observation set is being used, the labeling algorithm (Viterbi decoding) can be computationally expensive. In these cases, a “beam” search approximation is used (Bikel et al., 1999; Ratnaparkhi, 1996). While the “beam” search algorithm is not guaranteed to produce the optimal decoding, in practice, with an appropriate beam width, the algorithm discovers the optimal decoding in most cases.

In general, HMM approaches to NLP perform similarly or better than state of the art. (Brants, 2000) shows that an HMM approach to POS tagging performs similarly with state-of-art (Ratnaparkhi, 1996) but runs faster. Using HMMs for name entity recognition and classifications outperforms hand written rule based approaches whenever a small amount of data is available (Bikel et al., 1999). (Zhou & Su, 2002) HMM based name entity recognizer outperforms best performers in the MUC-6 and MUC-7 data. The HMM used in (Vogel et al., 1996) for word alignment produces better alignment than previous approaches for certain type of sentence pairs. Finally, (Barzilay & Lee, 2004) shows that their HMM-based content structure discovery model improves the state-of-art on two related tasks (information ordering and extractive summarization).

## References

- Barzilay, R., & Lee, L. (2004). *Catching the Drift: Probabilistic Content Models, with Applications to Generation and Summarization*. Paper presented at the HLT-NAACL.
- Berger, A. L., Pietra, S. A. D., & Pietra, V. J. D. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1), 39-71.
- Bikel, D. M., Schwartz, R., & Weischedel, R. M. (1999). An Algorithm that Learns What's in a Name. *Machine Learning (Special Issue on NLP)*.
- Brants, T. (2000). *TnT - A Statistical Part-of-Speech Tagger*. Paper presented at the ANLP.
- Chao, G., & Dyer, M. G. (2000). *Word Sense Disambiguation of Adjectives Using Probabilistic Networks*. Paper presented at the COLING.
- Chao, G., & Dyer, M. G. (2002). *Maximum Entropy Models for Word Sense Disambiguation*. Paper presented at the COLING.
- Chen, S. F., & Goodman, J. (1996). *An Empirical Study of Smoothing Techniques for Language*. Paper presented at the ACL.
- Chen, S. F., & Rosenfeld, R. (1999). *A Gaussian prior for smoothing maximum entropy models*: Technical Report CMUCS -99-108, Carnegie Mellon University.
- Ciaramita, M., & Johnson, M. (2000). *Explaining away ambiguity: Learning verb selectional preference with Bayesian networks*. Paper presented at the COLING.
- Darroch, J. N., & Ratcliff, D. (1972). Generalized Iterative Scaling for Log-linear Models. *Annals of Mathematical Statistics*, 43, 1470-1480.
- Galley, M., McKeown, K., Hirschberg, J., & Shriberg, E. (2004). *Identifying Agreement and Disagreement in Conversational Speech: Use of Bayesian Networks to Model Pragmatic Dependencies*. Paper presented at the ACL.
- Horvitz, E., & Paek, T. (1999). *A Computational Architecture for Conversation*. Paper presented at the User Modeling.

- Keizer, S., Akker, R. o. d., & Nijholt, A. (2002). *Dialogue Act Recognition with Bayesian Networks for Dutch Dialogue*. Paper presented at the SIGDial.
- Law, H. H.-C., & Chan, C. (1996). *N-th Order Ergodic Multigram HMM for Modeling of Languages without Marked Word Boundaries*. Paper presented at the COLING.
- Narayanan, S., & Jurafsky, D. (1998). *Bayesian models of human sentence processing*. Paper presented at the CogSci.
- Och, F., & Ney, H. (2002). *Discriminative training and maximum entropy models for statistical machine translation*. Paper presented at the ACL.
- Pakhomov, S. (2002). *Semi-Supervised Maximum Entropy Based Approach to Acronym and Abbreviation Normalization in Medical Texts*. Paper presented at the ACL.
- Rabiner, L. R. (1989). A tutorial on Hidden Markov Models and selected applications in speech recognition. *IEEE*, 77(2).
- Ramakrishnan, G., Jadhav, A., Joshi, A., Chakrabarti, S., & Bhattacharyya, P. (2003). *Question Answering via Bayesian Inference on Lexical Relations*. Paper presented at the ACL Workshop on Multilingual Summarization and Question Answering.
- Ratnaparkhi, A. (1996). *A maximum entropy model for part-of-speech tagging*. Paper presented at the Conference on Empirical Methods in Natural Language Processing (EMNLP).
- Reynar, J. C., & Ratnaparkhi, A. (1997). *A Maximum Entropy Approach to Identifying Sentence Boundaries*. Paper presented at the Conference on Applied Natural Language Processing (ANLP).
- Rosenfeld, R. (1994). *Adaptive Statistical Language Modeling: A Maximum Entropy Approach*. Unpublished Chapter 4 and 5.
- Rosenfeld, R. (1997). *A whole sentence maximum entropy language model*. Paper presented at the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU).
- Sato, K., & Nakanishi, M. (1998). *Maximum Entropy Model Learning of the Translation Rule*. Paper presented at the ACL.
- Skut, W., & Brants, T. (1998). *Chunk Tagger - Statistical Recognition of Noun Phrases*. Paper presented at the ESSLLI.
- Vogel, S., Ney, H., & Tillmann, C. (1996). *HMM-Based Word Alignment in Statistical Translation*. Paper presented at the COLING.
- Wai, C., Meng, H. M., & Pieraccini, R. (2001). *Scalability and Portability of a Belief Network-based Dialog Model for Different Application Domains*. Paper presented at the HLT.
- Zhou, G., & Su, J. (2002). *Named Entity Recognition using an HMM-based Chunk Tagger*. Paper presented at the ACL.
- Zukerman, I., McConachy, R., & Korb, K. (1998). *Bayesian Reasoning in an Abductive Mechanism for Argument Generation and Analysis*. Paper presented at the AAAI.
- Zukerman, I., McConachy, R., & Korb, K. (1999). *Exploratory interaction with a Bayesian argumentation system*. Paper presented at the IJCAI.