Methods and Theories for Large-scale Structured Prediction in Natural Language Processing

Shuming Ma
shumingma@pku.edu.cn

Abstract

This is a draft of a survey for large-scale structured prediction in natural language processing. Large-scale structured prediction is a important problem in language processing. In this paper, we introduce the significant theories and technologies for large-scale structured learning in recent years. Some important traditional models are Conditional Random Field, Structured Perceptron and Probabilistic Perceptron. These models have their own advantages in speed and performance. to reduce the labeling work of traditional models in large-scale leaning, latent variable models are used to replace the accurate labeled data. Recently, neural networks are widely used. They reduce the feature extraction work and improve the performance of large-scale structured learning. Finally, regularized models, especially structure regularization, can greatly reduce the overfitting risk of models.

1 Introduction

Language processing is a important part of Artificial Intelligence. As one of the core technologies of man-machine interface, Language Processing include two parts: language understanding and language generation. The former part is input and the later part is output. They act on the communication of humans and computers together.

structured learning is a important method in language processing. Language itself has complicated structures. How to recognize the hidden structures in these linear sequences is a important problem to explore. If language processing could recognize the various structure accurately in language, such like phrase structurerelation dependency, semantic role relations, language processing will benefit a lot.

However, the complexity and diversity of language greatly increase the difficulty of structured learning. In order to adapt to this characteristic of language, the use of large-scale corpus and large-scale model has become a hurdle for language processing, enhancement and performance enhancement.

Large scale structured learning, in practice, does bring about a significant improvement in its effectiveness. However, its slow learning speed limits its

application in a certain extent. The current large-scale structured learning involves two main points. First, to improve the learning speed, which makes the using of more large-scale data possible; Second, to maintain the original promotion effect in the acceptable range .

After a few years of development, large-scale structured learning has formed a complete set of theories, and a batch of practical verification techniques have emerged. This paper attempts to sort out the existing results and introduce existing solutions for different types of models and overfitting problems that caused by large-scale learning.

2 To Solve Structured Problems in Traditional Models

2.1 Conditional Random Fields

In traditional language processing, the model proposed by [9] proposed CRFs (Conditional Random Fields, CRFs) plays an important role. As a global probability model, its core is to maximize the conditional probability of correct output given input:

$$p(y|x,\theta) = \frac{1}{z(x,\theta)} exp(\sum_k \theta_k f_k(y,x))$$

$$Z(x,\theta) = \sum_{y'} exp(\sum_k \theta_k f_k(y',x))$$

x denote the input sequence, y denote the output sequence, y' denote all possible output sequence, θ denote the parameters(weights), f denotes the feature template function, Z denote the normalized function.

CRF learning methods are generally based on gradient maximum likelihood learning. Since the CRF predicts a global structure every time, rather than a local label, it is necessary to search for the optimal structure from the input according to the model parameters at prediction stage. The process is called the decoding process and is often based on dynamic programming algorithm (Viterbi Algorithm), in order to avoid the extremely high complexity of enumeration traversal.

2.2 Structured Perceptron

Nevertheless, the training efficiency of CRF is still low. [2] proposed CRF (Structured Perceptron) algorithm, which solves the problem of training speed to some extent. In theory, it ensure that algorithm is convergent if the data can be divided. This algorithm avoids the gradient calculation in CRF, and updates the parameters only for the input that is predicted incorrectly. In practical applications, structured perceptrons have a huge

speed advantage over CRF, but their effectiveness is hardly comparable to that of CRF. Based on the idea of structured perceptrons, there are a number of improved versions that appeared. They aim to further improve the speed or further enhance the effect.

According to the characteristics that structured perceptron is still using precise search Vitby algorithm in decoding, researchers have proposed using greedy search (greedy search) or column search (beam search) and other non precise search scheme, to further enhance the learning and speed of prediction.

The policy of discretization parameter updating may be too rough and simple. There are many quick strategies. Some well-known strategies are earlier (Early Update) update algorithm proposed by [3], Margin Infused Relaxed Algorithm (MIRA) proposed by [4]. Max-violation algorithm proposed by [8]. However, these improvements, despite their speed advantage over CRF, are hardly comparable to CRF's.

2.3 Probabilistic Perceptron

The traditional model is difficult to satisfy both speed and effect. CRF has good performance but is slow to train. Structured Perceptron has fast speed but the effect is relatively low, while other models such as MIRA and k-best MIRA are between the two situations. In the large-scale structured learning, the speed and effect of the model are important indicators of model's applicability. Recently, the Probabilistic Perceptron proposed by [15] (Prob-

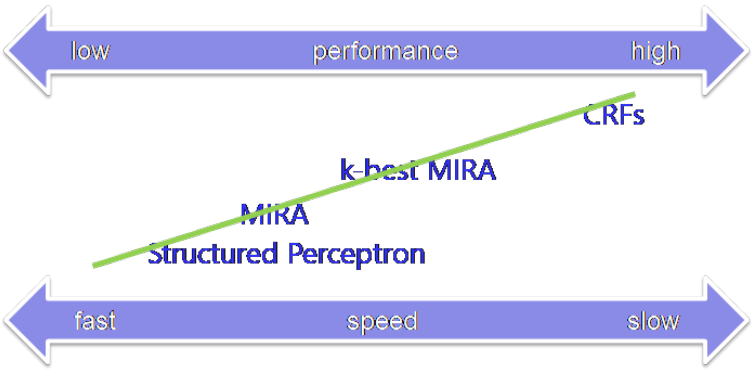

Figure 1: Compromise between effect and speed among different models

abilistic Perceptron SAPO) model is a good solution which balances speed and effect. In the tasks of actual large-scale structured prediction, it achieves effect that comparable with the CRF effect and speed that comparable with structured perceptron. The core idea of the model is to introduce the probabilistic information of suboptimal prediction in structured perceptron. The

model also ensures that learning is convergent in theory. The experimental results of the model in terms of word class prediction and phrase segmentation are not inferior to CRF, and are much higher than those of structured perceptron, MIRA and other algorithms.

In addition, the probabilistic perceptron has pretty good performance in speed.

Compared to the MIRA or structured perceptron, another feature of the model is that the complexity of feature weight is not improved, which means the learning essence is similar to that of CRF.

2.4 Conclusions

In summary, in the traditional model, CRF has a great effect, but its slow learning speed is difficult to adapt to large-scale structured learning tasks. Structured Perceptron and its improved version have a larger advantage in speed, but somewhat lacking in effect.Probabilistic Perceptron (SAPO) unities the speed and effect, is an ideal scheme for large-scale structured learning.

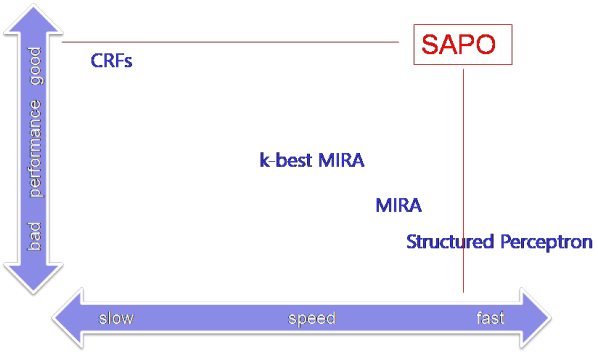

Figure 2: Probabilistic Perceptron

3 Using Latent Variable Models to Reduce Labeling Work

Traditional models require a large amount of labeled data when solving large-scale structured learning problems. In practice, data labeling faces many challenges. Some tasks are difficult to mark and lack sufficient data annotation, which results in poor training effect. The existing annotation data will affect the training effect of the model because the annotation is not accurate enough. In order to solve the problem of data annotation, some scholars have proposed latent variable model. The latent variable model

replaces the accurate annotation data by introducing latent variables, so that the annotation data does not need to have fine granularity, thus reducing the workload of data annotation and improving the effect of the model.

3.1 The Importance of Latent Variable Models

Latent variable model has been widely used in natural language processing. [10] and [11] use the latent variable model for syntactic analysis to learn more elaborate grammar. [7] use it in dependency parsing based on transition. They transform the transition state into a collection of transitions of partial annotations and achieve better effect.[5] transform the problem into query and then transform the query into answer Q and A system. Because it is difficult to obtain accurate query, they treat query as a hidden variable to deal with. [23] treats intention as a hidden variable to deal with in the multi-intention speech recognition and achieves good results.

3.2 Probabilistic Structured Perceptron

A commonly used latent variable model is the latent variable conditional random field (Fig. 9). latent variable conditional random field increases the number of latent variables between the input sequence and the output tag sequence to learn the potential information.

However, the training of latent variable conditional field is very slow, and it may take several weeks to converge, which greatly reduces the availability of latent variable random conditional field. In structured learning, slowness means that the model is difficult to deal with large-scale data. Since the training of Structured Perceptron is faster than the conditional random field, [19] and [18] proposed a solution to transform the latent variable conditional random field into a latent variable structured perceptron.

The latent variable structured perceptron only retains the latent variable sequence which results in real tag sequence, rather than searching for all possible sequences like the latent variable conditional random field. In this way, it can greatly reduce the time required for training. In addition, the latent variable structured perceptron can also be proved to be convergent in theory, which provides a guarantee for the latent variable structured perceptron in theory . Experiments also show that the latent variable structured perceptron can achieve better results than that of latent variable conditional random field.

3.3 The Application of Latent Variable Structured Perceptron

Latent variable structured perceptron has been applied in different tasks and it achieves good performance. [5] proposed an OQA model based on the latent variable structured perceptron, and achieved good experimental results

in the Q and A system task. [23] treats multi-intention as latent variables in the voice classification task and gets a good effect by using of latent variable structured perceptron. [24] used the latent variable structured perceptron to improve the semantic analysis task by using the mixed tree as latent variables. [7] and [6] also achieve a good effect on the dependent parsing and the anaphora disambiguation task by using the latent variable structured perceptron.

3.4 Conclusions

Hidden variable model can reduce the workload of data annotation and improve the effect of traditional models. The training of Latent variable CRFs is slow. While the latent variable structured perceptron can improve both the training speed and effect, and ensure the convergence in theory. Thus it achieves good experimental results in the question answering system, speech classification, syntactic parsing, semantic analysis and anaphora disambiguation tasks .

4 Using Neural Networks to Reduce Feature Extraction Work

In the evolution of models, the latent variable model alleviates the heavy workload of manually annotated corpus, but the extraction of model features still requires manual assignment. The precise design of feature templates often requires linguistic knowledge, and different tasks require targeted feature templates. In large-scale structured learning, the actual features tend to be millions, and the complexity of them can be seen.

The emergence of neural models solves the problem of artificial feature extraction. Neural model, that is, neural network based language processing model, has become one of the focuses in the field of language processing. The neural model replaces the sparse features of the original model by real self-learning intensive features, simplifies the process of language processing, and also makes some tasks which are difficult to extract features feasible.

4.1 Neural Networks

In short, neural network has three categories: Feed-Forward Neural Networks, Convolutional Neural Network (CNN) and Recurrent Neural Networks (RNN). Because of the limited space, the training of neural networks and some other details are not the focus of this paper.

Because of the linear "timing" characteristics of language, in practice, the neural model is dominated by recurrent neural network. Recurrent neural network, as its name suggests, is a neural network that neurons appear

repeatedly. It was first used in speech analysis to predict linear structure by modeling temporal information.

However, common recurrent neural networks do not perform very well in language processing tasks. The main problem is the gradient vanishing / explosion phenomenon, which makes it difficult to capture the long distance dependencies in languages for neural networks. This problem is solved by the Long Short-Term (Memory) model to some extent. The model consists of a continuous linear memory module which captures long-distance dependencies. The gate controls the modify of the memory module.

The complexity of LSTM model makes it very powerful for fitting. Complex models require large-scale data to learn, and large-scale data, in turn, enhance their ability. LSTM has achieved significant improvements over traditional models or latent variable models in many tasks. For example, the most well-known sequence to the sequence model, has made important breakthroughs in the machine translation, machine summarization, emotional analysis, dialog generation and other difficult areas. It is amazing that such a simple structure can achieve such an effect. amazing 13).

For large structured learning tasks, the main problem with LSTM is that the model is too large, resulting in very slow computation. A neural Machine Translation model may take several weeks to train even equipped with GPU acceleration. Its high cost of learning time limits its use in more comprehensive tasks. Although the main solutions of speeding up neural model learning is still using GPU cluster computing. This scheme can not solve the problem of high computational complexity fundamentally.

The attempts to simplify the structure of the LSTM model have few progress. The only notable thing is the Gated Recurrent Unit(GRU), proposed by the [1], which combines the forger gate and input gate of the structure and makes structural deformations.

The structure only reduces only one gate and one nonlinear operation compared to the LSTM cell structure. Practice has also shown that although the model is equivalent to LSTM in effect, its speeding is not very obvious. Further research is necessary for speeding up.

Another problem with the application of neural models is that the original CPU computing resources are partially idle. The advantages of GPU in matrix computation are fully exploited when large-scale neural model learning is used. While CPU is in a relatively inferior position. How to use its previous computing resources is also a question worth thinking about.

4.2 Asynchronous Parallel Training of Neural Network Models

Although the parallel efficiency of CPU is not high when large scale matrix computing is needed in mini-batch, the speeds of CPU and GPU are neck and neck in online training which means training only one sample

at a time. Therefore, the parallel version of online training is very attractive [21, 17]. But in practice, the synchronous operation to maintain accuracy severely drags parallel acceleration. The asynchronous parallel training (Asynchronous, Parallel, Learning) scheme proposed by [16] solved this problem.

Algorithm 1 *AsynGrad*: Asynchronous Parallel Learning with Gradient Error

Input: model weights $\boldsymbol{w}$, training set S of m samples
Run k threads in parallel with share memory, and procedure of each thread is as follows:
repeat
  Get a sample $\boldsymbol{z}$ uniformly at random from S
  Get the update term $\boldsymbol{s_z}(\boldsymbol{w})$, which is computed as $\nabla f_{\boldsymbol{z}}(\boldsymbol{w})$ but usually contains error
  Update $\boldsymbol{w}$ such that $\boldsymbol{w} \leftarrow \boldsymbol{w} - \gamma \boldsymbol{s_z}(\boldsymbol{w})$
until Convergence
return $\boldsymbol{w}$

Figure 3: AsynGrad algorithm.

Asynchronous training algorithm has been applied in the traditional sparse feature models, and the convergence of it has been proved in theory. But the neural model belongs to intensive model. Its operation mode in learning process significantly different from that of sparse feature model.

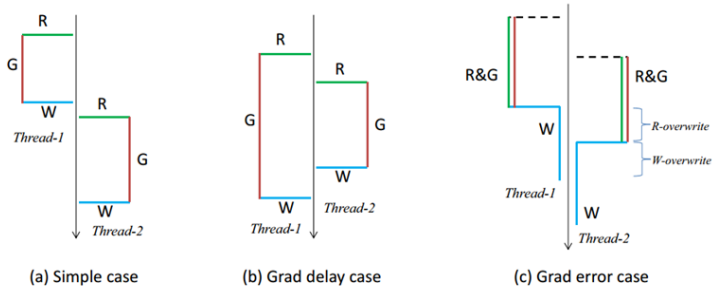

Figure 4: Different cases.

It is difficult to avoid the gradient error in asynchronous neural model learning, the contribution of AsynGrad is proved that in theory, asynchronous learning will still converge to a point that is near the optimal point in the final stage despite gradient error, as long as the gradient error is controlled in a certain range.

The application in practice also verifies the correctness of the algorithm. It is an alternative learning scheme that utilizes existing CPU resources for large-scale structured learning.

4.3 Conclusions

Although neural models are applied to language processing before too long, their validity has been confirmed by [22]. However, its large computational complexity makes it difficult to apply it to large-scale structured learning tasks. The current solutions do not fundamentally solve problem: the structure that is partially streamlined do not significantly reduce the learning time, parallel learning is based on the complicated computation. Thus, there are still many practical problems to solve for neural models in large-scale structured learning [20].

5 Using Regularization to Avoid Overfitting

The problem of overfitting is often encountered in large scale structured prediction. Overfitting means that the error rate of the model in training set is very low but is high in test set. Overfitting is caused by the complex structure of model. To solve the problem, we need to introduce penalty items into the parameters to reduce the complexity of the model.

5.1 Weight Regularization

one of the widely used regularization methods is to add the L1 penalty item or the L2 penalty term in the loss function:

$$\min_{w} loss(x, y, x) + \lambda regularizier(w)$$

where $loss(x, y, w)$ denotes the original loss function, and $regularizier(w)$ denotes the weighting regularization term while lambda is the regularization factor. If the regularization term is the L1 regular term, then

$$regularizer(w) = \parallel w \parallel$$

If the regularization term is the L2 regular term, then

$$regularizer(w) = \frac{1}{2} \parallel w \parallel_2$$

5.2 Structure Regularization

One of the reasons for overfitting is that the complexity of the model is too high. To solve the overfitting problem, the complexity of the model needs to be reduced. [13] proposed structure regularization to solve over fitting problems. In large-scale structured learning problems, the structure of many models is very complex, and the idea of structure regularization is to decompose the complex model structure into simple structure. Training of

structure regularization is fast and easy to implement. Theoretical guarantee is also offered.

The graph is a schematic diagram of structure regularization. It can find the proper model complexity by simply decomposing the structure, and it is very simple to implement. It can be regarded as the preprocessing of training data.

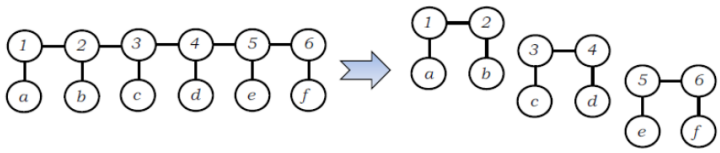

Figure 5: Structure regularization.

A problem with structure regularization is whether the decomposed structure will cause loss of long distance features. There are two solutions of this problem. One is that extracting features before the structure decomposition. Another is that only decomposing the output sequence structure to maintain input sequence information, so that we can reduce the complexity of the model based on the features of original data.

Since the solutions aim to solve the overfitting problem, the structure regularization only needs to decompose the structure of training data, and does not need to decompose the structure of test data. Because it is to solve the overfitting problem in the training process.

Through theoretical analysis, it is known that models trained by complex structures yield low empirical risks, but can lead to high overfitting risk. The models trained by simple structure have high empirical risk, but the risk of overfitting is relatively low. So we need to find a suitable structure complexity to balance the relationship between the two. Such a suitable structure can have good accuracy in training and is not easy to overfitting, and can achieve high accuracy at testing stage. The simple structure after decomposition can not only improve the stability, but also has better generalization ability.

Structure regularization has several significant advantages. First, due to simple structures to be trained, the training cost, or the number of iterations required for convergence is also less. And the objective function maintains convex after regularization if the objective function is convex. Second, the structure regularization and weight regularization do not conflict, a model can apply both structure regularization and weight regularization, as far as possible to reduce the risk of over fitting. Finally, the structure regularization and model itself are independent of each other. Structure regularization can be treated as a pretreatment process.

The experiments also prove that the structure regularization can achieve

better results when applied in conditional random field and structure perceptron [14]. And the training speed is faster.

5.3 Dropout

Dropout is a technique proposed by [12] to avoid overfitting of neural network models. In the training stage, some neurons do not participate in forward propagation and backward propagation. It reduces the complexity of the model by reducing the parameters and the risk of overfitting is reduced too. In addition, it can also reduce training time.

5.4 Conclusions

Overfitting is one of the most common problems in large-scale structured learning. In order to reduce the risk of overfitting, it is necessary to reduce the complexity of the model through some techniques. Weight regularization, structure regularization and dropout are popular methods to solve overfitting problems. Weight regularization avoids overfitting by introducing penalty term to weight; structure regularization directly reduces the complexity of the model; dropout avoids overfitting problem by reducing neural network parameters randomly. These technologies do not interfere each other and can be combined to reduce the risk of overfitting.

6 Conclusions

In order to solve the problems of large scale structured learning for language processing, a number of models have been proposed in recent years. The traditional models includes conditional random field, structured perceptron and probabilistic perceptron, which can solve the problems of structured prediction to some extent. However, these models requires annotation and features that need to be precessed artificially. To reduce the artificial work, latent variable models are applied to tasks where data need to be labeled but marks are difficult. In order to reduce features extracted artificially, neural network models are used to learn potential features automatically in data. Some regularization methods, including weight regularization, structure regularization, and dropout, are used to solve the overfitting problem in large-scale data. These models and methods constitute the basic framework of large-scale structured learning.

In future, structured learning models and technologies will be used more widely in the Natural Language Processing field due to the structural nature of natural languages. The tasks of Machine Translation, syntax analysis, question answering system and text summarization can not be separated from models or methods of structured learning. Solving the problems of structured learning will be the only way to understand natural languages.

References

[1] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[2] Michael Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 1–8. Association for Computational Linguistics, 2002.

[3] Michael Collins and Brian Roark. Incremental parsing with the perceptron algorithm. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 111. Association for Computational Linguistics, 2004.

[4] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7(Mar):551–585, 2006.

[5] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1156–1165. ACM, 2014.

[6] Eraldo Rezende Fernandes, Cícero Nogueira dos Santos, and Ruy Luiz Milidiú. Latent trees for coreference resolution. *Computational Linguistics*, 2014.

[7] Matthew Honnibal and Mark Johnson. Joint incremental disfluency detection and dependency parsing. *Transactions of the Association for Computational Linguistics*, 2:131–142, 2014.

[8] Liang Huang, Suphan Fayong, and Yang Guo. Structured perceptron with inexact search. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–151. Association for Computational Linguistics, 2012.

[9] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[10] Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. Probabilistic cfg with latent annotations. In *Proceedings of the 43rd annual meeting on Association for Computational Linguistics*, pages 75–82. Association for Computational Linguistics, 2005.

[11] Slav Petrov and Dan Klein. Sparse multi-scale grammars for discriminative latent variable parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 867–876. Association for Computational Linguistics, 2008.

[12] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014.

[13] Xu Sun. Structure regularization for structured prediction. In *Advances in Neural Information Processing Systems*, pages 2402–2410, 2014.

[14] Xu Sun. Structure regularization for structured prediction: Theories and experiments. 2014.

[15] Xu Sun. Towards shockingly easy structured classification: A search-based probabilistic online learning framework. *arXiv preprint arXiv:1503.08381*, 2015.

[16] Xu Sun. Asynchronous parallel learning for neural networks and structured models with dense features. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 192–202, 2016.

[17] Xu Sun and Shuming Ma. Lock-free parallel perceptron for graph-based dependency parsing. *CoRR*, abs/1703.00782.

[18] Xu Sun, Takuya Matsuzaki, and Wenjie Li. Latent structured perceptrons for large-scale learning with hidden information. *IEEE Transactions on Knowledge and Data Engineering*, 25(9):2063–2075, 2013.

[19] Xu Sun, Louis-Philippe Morency, Daisuke Okanohara, and Jun'ichi Tsujii. Modeling latent-dynamic in shallow parsing: a latent conditional model with improved inference. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 841–848. Association for Computational Linguistics, 2008.

[20] Xu Sun, Xuancheng Ren, Shuming Ma, and Houfeng Wang. meprop: Sparsified back propagation for accelerated deep learning with reduced overfitting. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 3299–3308, 2017.

[21] Xu Sun, Houfeng Wang, and Wenjie Li. Fast online training with frequency-adaptive learning rates for chinese word segmentation and new word detection. In *Proceedings of the 50th Annual Meeting of*

the Association for Computational Linguistics: Long Papers-Volume 1, pages 253–262. Association for Computational Linguistics, 2012.

[22] Jingjing Xu and Xu Sun. Dependency-based gated recursive neural network for chinese word segmentation. In *The 54th Annual Meeting of the Association for Computational Linguistics*, page 567, 2016.

[23] Puyang Xu and Ruhi Sarikaya. Exploiting shared information for multi-intent natural language sentence classification. In *INTERSPEECH*, pages 3785–3789, 2013.

[24] Junsheng Zhou, Juhong Xu, and Weiguang Qu. Efficient latent structural perceptron with hybrid trees for semantic parsing. In *IJCAI*, pages 2246–2253, 2013.