Addressing distribution shift at test time in pre-trained language models

Anonymous

Anonymous organization
Anonymous address
Anonymous email

Abstract

State-of-the-art pre-trained language models (PLMs) outperform other models when applied to the majority of natural language processing tasks. However, one challenge with PLMs is how to prevent performance degradation under distribution shift, a phenomenon that occurs when data at test-time does not come from the same distribution as the source training set. Equally as challenging is the task of obtaining labels in real time due to issues like long-labeling feedback loops. The lack of adequate methods that address the aforementioned challenges constitutes the need that is addressed in this article – an approach that will continuously adapt the PLM to a distinct distribution. Unsupervised test-time domain adaptation adapts a source model to an unseen as well as unlabelled target domain. While some techniques such as unsupervised data augmentation are able to adapt models in different ways, they have only been sparsely studied for addressing the distribution shift problem. In this work, we present an approach (MEMO-CL) that improves the performance of PLMs at test-time under distribution shift. Our approach takes advantage of the latest unsupervised techniques both in data augmentation and adaptation to minimize the entropy of the PLM's output distribution on a batch of augmented samples from a single test point. The technique introduced is unsupervised, domain-agnostic, easy to implement, and requires no additional data. Our experiments result in a 3% improvement over other baselines on a strong text adaptation benchmark.

In the field of machine learning, the notion that a model will remain accurate over long periods of time is often times accepted as valid. Nonetheless, multiple studies have shown that models do not necessarily perform as well over time when evaluated (Quinonero-Candela et al. 2008; Sugiyama and Kawanabe 2012). This phenomenon of models performing worse due to a change in data is known as *distribution shift*. Distribution shift refers to the change in the underlying semantics of the data that is used to evaluate a model. In the language domain, the ever-changing nature of language as its spoken or written can prove to be a key contributing factor behind the distribution shift. Therefore, this degradation over time has been found to be prevalent in pre-trained language models (PLMs) as well (Lazaridou et al. 2021). In an attempt to resolve degradation, one could attempt to re-train a PLM (Kim et al. 2022), however, re-training is not only cost prohibitive but could also lead to catastrophic failures (Bender et al. 2021). Adapting at test-time does not require re-training, however, does require methods to have low runtime complexity. Several works have attempted to address the adaptation to recent data, albeit without addressing the application of those methods at test-time.

There are a plethora of augmentation methods that improve performance during the training process. Nonetheless, ones that address adaptation during the test-time are less studied, when distribution shift often occurs (Wiles et al. 2022). Textual augmentation methods range from naive-character perturbations (Wang and Yang 2015) and synonym replacements (Ganitkevitch, Van Durme, and Callison-Burch 2013) to vector space models that use context (Kobayashi 2018; Thakur et al. 2021). One approach that can be used during testing is called test-time augmentation (TTA). TTA aggregates predictions on a batch of augmented samples to form a final prediction (Molchanov et al. 2020; Kim, Kim, and Kim 2020; Shanmugam et al. 2021). (Lu et al. 2022) used TTA to improve accuracy in toxicity detection by averaging the predictions of the classifier on a batch of augmented samples generated from a single inference sample. Furthermore, (Yan, Guo, and Yang 2021) used confidence estimates to select high-quality augmentations. These TTA approaches increase robustness per test sample in isolation, whereas in the case of distributional shifts, the underlying population itself changes. An ideal technique would not only increase the per-sample robustness of PLM but the entirely shifted distribution as well.

Adapting pre-trained models to a continual stream of distinct data is known as *continual learning* (CL) (Pfülb 2022). On the one hand, CL algorithms are successful in presence of labels (Zhuang et al. 2021; Chawla, Singh, and Drori 2021), but on the other hand are sub-optimal in the absence of labeled data, especially under distribution shift. Since label acquisition can be difficult at test-time, some have used unsupervised methods (Ma et al. 2019; Wu, Yue, and Sangiovanni-Vincentelli 2021; Mishra, Saenko, and Saligrama 2021), while others used self-supervision (Sohn et al. 2020; Sun et al. 2020; Pérez-Carrasco, Protopapas, and Cabrera-Vives 2021; Chen et al. 2022) to adapt a model with unlabelled data. (Machireddy et al. 2022; Jin et al. 2022; Cossu et al. 2022) found that continuing to pre-train models using the original unsupervised objective miti-

gates forgetting as well as adapts the model to the latest data. However, their methods are data-intensive, whereas at test time only a single sample is received. Therefore, we frame this as a test-time adaptation problem where the model needs to be continually adapted from an in-domain source to an out-domain target with one unlabelled test point.

In this work, we introduce a technique named MEMO-CL based on the marginal entropy minimization over a single test sample (MEMO (Zhang, Levine, and Finn 2022)) and extend it to the CL on text paradigm. (Zhang, Levine, and Finn 2022) makes the case that domain adapting a model using TTA before taking the final prediction leads to increased robustness. Similarly, our proposed approach continually trains a model by doing TTA before adaptation. MEMO-CL encourages the PLM to predict similarly for semantically similar examples that are augmentations of a single data point. Our alternative method approaches test-time domain adaptation in a distinct manner from previous methods. First, it generates synthetic samples from a test sample. Second, it filters those samples using predictions from the base model. Finally, it adapts the model weights before using it for the final prediction. We compare MEMO-CL to an extensive set of baseline methods and find that it improves performance and robustness under distribution shift.

In order to better illustrate our method, we provide a more detailed description in the following sections. First, we expand on our methodology that adapts models to continually changing data at test-time. Next, we introduce how the quality of the corpus was further improved by selecting informative samples using a margin-based filtering approach. Then, we shed light on the type of data augmentations that MEMO-CL uses. Finally, we share the experiments performed and a discussion on their corresponding outcomes.

Methodology

In this section, we formally define the MEMO-CL approach. We first describe how our streaming augmentation approach is adapted to test-time scenarios using what we call *unsupervised test-time adaptation*. We then cover a method of removing noise produced from the augmentations called *semantic margin-based filtering*. Finally, we show how we use data augmentation to provide samples for MEMO-CL. Algorithm 1 depicts the end-to-end approach.

Unsupervised test-time adaptation (UTTA)

Algorithm 1 (lines 14–15) illustrates how domain adaptation takes place using MEMO-CL. More formally, given an inference sample $x \in \mathcal{X}$ used to predict label $\hat{y} \in \mathcal{Y}$ using a model $f(x;\theta) : \theta \in \Theta$. To d so, from a uniform distribution $\mathcal{U}(\mathcal{A})$ of augmentation functions $a \in \mathcal{A}$, a batch of $N \in \mathbb{N}$ augmented samples $\tilde{x} \leftarrow a_i(x)\, \forall\, i \in [1, N]$ are generated. Therefore, the expectation $\mathbb{E}$ of the original model's conditional output distribution $p_\theta(y|\tilde{x})$ being consistent across augmentations $\tilde{x}$ is given in Equation 1 as follows:

$$p_\theta(y|x) \triangleq \mathbb{E}_{\mathcal{U}(\mathcal{A})}[p_\theta(y|a(x))] \approx \frac{1}{N}\sum_{n=1}^{N} p_\theta(y|\tilde{x}_n). \quad (1)$$

Algorithm 1: Test-time adaptation via MEMO-CL algorithm

Input: input $x \in \mathcal{X}$, model $f(x;\theta) : \theta \in \Theta$
Parameter: margin δ, number of augmentations $N \in \mathbb{N}$, model weights $\theta \in \Theta$, learning rate $\eta \in \mathbb{R}$, optimizer $\mathcal{O}$
Output: predicted label $\hat{y} \in \mathcal{Y}$

1: Let $i \leftarrow 0, p_\theta \leftarrow 0, \tilde{X} \leftarrow \emptyset$
2: **while** $i < N$ **do**
3: $\quad a \sim \mathcal{U}(\mathcal{A})$ ▷ sample augmentation function (DA)
4: $\quad \tilde{x} \leftarrow a(x)$ ▷ augment x (DA)
5: $\quad \tilde{p_\theta} \leftarrow f(\tilde{x}|\theta)$ ▷ predict using model
6: $\quad$ **if** $\tilde{p_\theta} - \delta < \tilde{p_\theta} < \tilde{p_\theta} + \delta$ **then** ▷ Eq. 3 (SMF)
7: $\quad\quad \tilde{X} := \tilde{X} \cup \{\tilde{x}\}$
8: $\quad\quad p_\theta := p_\theta + \tilde{p_\theta}$
9: $\quad\quad i := i + 1$
10: $\quad$ **else**
11: $\quad\quad$ continue
12: $\quad$ **end if**
13: **end while**
14: $\ell \leftarrow H(p_\theta/N)$ ▷ compute loss using Eq. 2 (UTTA)
15: $\theta' \leftarrow \mathcal{O}(\theta, \eta, \ell)$ ▷ update model weights (UTTA)
16: **return** $\hat{y} \triangleq \arg\max_{y \in \mathcal{Y}} f(x|\theta')$

One of the main goals of MEMO-CL is to adapt model weights so that the adapted model performs well on unlabelled target data without sacrificing performance. This is done in a streaming manner which modifies the original work on MEMO (Zhang, Levine, and Finn 2022) that limited domain adaptation to per sample basis. The MEMO-CL approach hence assumes that the distribution shift is not limited to one point in time, in turn covering the entire test-time data set. MEMO-CL rewards similar predictions from the model that are invariant to perturbations via augmentations. The idea contrasts the standard form of the cross-entropy loss method which penalizes *confident* yet incorrect predictions. It does so by minimizing the entropy H of marginal output distribution using loss ℓ as follows in Equation 2:

$$H(p_\theta(y|x)) \triangleq -\sum_{y \in \mathcal{Y}} p_\theta(y|x) \log p_\theta(y|x) \quad (2)$$

Semantic margin-based filtering (SMF)

In Algorithm 1 (lines 6–13), we demonstrate how MEMO-CL handles erroneous augmentations (noisy examples) produced by UTTA. Noisy examples are filtered by keeping only those examples considered crucial for semantic purposes. More formally, semantics preserving augmented sample pool $\tilde{x}$ that are within margin δ of the prediction probability p_θ of a given sample are kept using an indicator function $\mathbb{1}_{SMF}$ defined in Equation 3:

$$\mathbb{1}_{SMF}(p, \delta) := \begin{cases} 1, & \text{if } p - \delta < p < p + \delta \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Combining Equation 3 with Equation 2 for adaptation, the final loss function can thus be written as in Equation 4:

$$\ell(x;\theta) := \mathbb{1}_{SMF}(p_\theta(y|x), \delta) H(p_\theta(y|x)) \quad (4)$$

Data augmentation (DA)

Another goal of the MEMO-CL approach is to preserve the semantic meaning of the resulting augmented samples. To this end, we follow previous work on contextual augmentation approaches that provide highly-diverse examples while preserving semantics (Feng et al. 2021). In their work, diversity is defined as the number of semantically-similar modifications performed on the text. Similarly, MEMO-CL uses the following three state-of-the-art context-based augmentation techniques $\mathcal{A}$ at test-time:

1. Synonym replacement using a database of frequent paraphrases (Ganitkevitch, Van Durme, and Callison-Burch 2013).
2. Synonym replacement by words having similar (measured via cosine distance) dense representations using word2vec (Mikolov et al. 2013).
3. Use a standard pre-trained language model to substitute or insert phrases (Kobayashi 2018; Kumar, Choudhary, and Cho 2021).

A uniform distribution $\mathcal{U}(\mathcal{A})$ from all of the above-listed augmentation techniques $\mathcal{A}$ is created to be used in the MEMO-CL, as defined in Algorithm 1 (lines 3–4).

Experimental settings

In this section, an explanation follows of the experiments performed. The model was adapted with a learning rate η of 10e-5 on a machine with 4 NVIDIA V100 GPU, 24 CPU, and 448GB RAM (*Azure* NC24v3). Adapting the model takes 10 seconds per sample with $N = 20$ augmentations.

Dataset. To evaluate our method, we select a dataset that has a significant distributional shift between train and test distribution. The standard for this is WILDS-CivilComments dataset (Koh et al. 2020) which is a modification of the dataset by (Borkan et al. 2019). This dataset contains 269,038 train and 133,782 test samples along with metadata on belonging to one or more of the 8 sensitive groups. The inputs are sequences with their corresponding binary labels of whether the input is toxic or not.

Baseline. The baseline is identical to the one used by (Koh et al. 2020) which is a DistilBERT (Sanh et al. 2019) model fine-tuned on a binary classification task for toxicity detection on the WILDS-CivilComments dataset.

TTA. Apart from the baseline, we also compare MEMO-CL with numerous TTA methods. Those are split between *hard* (predicted label) and *soft* (predicted logit) methods. There are numerous ways in which predictions can be aggregated, the primary ones being (1) *majority-voting* uses mode of highest predicted class (Wu, Yue, and Sangiovanni-Vincentelli 2021) (2) *average* simply takes mean of all logits (Lu et al. 2022) (3) *class weighted* learns optimal weights from a pool of augmentations (Shanmugam et al. 2021).

MEMO-CL. To generate contextually augmented samples $\tilde{x}$, the default configuration of *nlpaug* library (Ma 2019) is used to uniformly sample from $a \sim \mathcal{U}(\mathcal{A})$ at test-time. Our method's performance is reported along with improvement achieved by using SMF ($\delta \leftarrow 0.1$).

Evaluation. The original metrics used by (Koh et al. 2020) included measuring overall *average accuracy* (AA) as well as *worst group accuracy* (WGA) among the 8 different marginalized groups. Additionally, the degree of loss in performance of the approach on the original dataset was measured using *correction to corruption ratio* (CCR). The corruptions are defined as the number of originally correct predictions that were flipped incorrectly, whereas corrections are defined as the number of originally incorrect predictions that were flipped correctly.

Results and conclusion

Experiments confirm our initial hypothesis of augmenting and then adapting before taking the final prediction. MEMO-CL outperformed baseline as well as TTA methods. Furthermore, by continually learning from incoming data, we notice an additional performance boost over baseline MEMO. The method reduces variance in accuracy not only by minimizing the entropy per sample but that of a stream of augmented samples, thus, increasing label efficiency. Further reduction in variance was achieved by margin-based filtering (SMF). The high variance of TTA can be attributed to the augmentations not being perfectly semantic preserving as found by (Lu et al. 2022). The proposed approach also exhibited the highest CCR, however, it was noticed as having higher variance compared to TTA. This adds value by accounting for the inherent noise in the dataset e.g. fixing corruptions by essentially flipping a higher number of corrupt samples.

Model	AA ↑	WGA ↑	CCR ↑
Baseline DistilBERT	92.3	53.7	
TTA majority hard-voting	- 0.6 (.4)	- 0.2 (.1)	0.91 (.04)
TTA majority soft-voting	+0.2 (.1)	+0.4 (.2)	1.06 (.05)
TTA average	+1.1 (.2)	+0.8 (.1)	1.11 (.03)
TTA class weighted	+1.4 (.5)	+0.6 (.3)	1.16 (.03)
MEMO	+1.6 (.2)	+0.9 (.1)	1.12 (.04)
MEMO-CL	+2.4 (.3)	+1.2 (.2)	1.19 (.09)
MEMO-CL + SMF	**+2.9** (.1)	**+1.6** (.1)	**1.21** (.11)

Table 1: Comparison of MEMO-CL with baseline and existing TTA approaches. The parentheses enclose the standard deviation from 5 runs. See evaluation for details of metrics.

In this work, a novel technique is presented for adapting PLMs to distributional shifts on the fly via augmentation followed by adaptation. Inspiration was taken from work done in unsupervised augmentation, adaptation, and test-time robustness. The technique is simple to implement, extensible to multi-class scenarios, domain-agnostic, and does not require any labeled data. Experiments show that the MEMO-CL method improves the average and worst-group accuracy over existing approaches.

One potential extension of the present work would be an experimental setup that evaluates the dexterity of the approach in handling not one but multiple simultaneous shifts. We also posit that learning which type of augmentations will be more suitable for adaptation proposed by (Cubuk et al. 2019) is an interesting direction for future work.

References

Bender, E. M.; Gebru, T.; McMillan-Major, A.; and Shmitchell, S. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? . In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, 610–623. New York, NY, USA: Association for Computing Machinery. ISBN 9781450383097.

Borkan, D.; Dixon, L.; Sorensen, J.; Thain, N.; and Vasserman, L. 2019. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion Proceedings of The 2019 World Wide Web Conference*.

Chawla, S.; Singh, N.; and Drori, I. 2021. Quantifying and Alleviating Distribution Shifts in Foundation Models on Review Classification.

Chen, B.; Jiang, J.; Wang, X.; Wan, P.; Wang, J.; and Long, M. 2022. Debiased Self-Training for Semi-Supervised Learning. ArXiv:2202.07136 [cs].

Cossu, A.; Tuytelaars, T.; Carta, A.; Passaro, L. C.; Lomonaco, V.; and Bacciu, D. 2022. Continual Pre-Training Mitigates Forgetting in Language and Vision. *ArXiv*.

Cubuk, E. D.; Zoph, B.; Mane, D.; Vasudevan, V.; and Le, Q. V. 2019. AutoAugment: Learning Augmentation Policies from Data. ArXiv:1805.09501 [cs, stat].

Feng, S. Y.; Gangal, V.; Wei, J.; Chandar, S.; Vosoughi, S.; Mitamura, T.; and Hovy, E. 2021. A Survey of Data Augmentation Approaches for NLP. ArXiv:2105.03075 [cs].

Ganitkevitch, J.; Van Durme, B.; and Callison-Burch, C. 2013. PPDB: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 758–764.

Jin, X.; Zhang, D.; Zhu, H.; Xiao, W.; Li, S.-W.; Wei, X.; Arnold, A.; and Ren, X. 2022. Lifelong Pretraining: Continually Adapting Language Models to Emerging Corpora. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, 1–16. virtual+Dublin: Association for Computational Linguistics.

Kim, D.; Wang, K.; Sclaroff, S.; and Saenko, K. 2022. A Broad Study of Pre-training for Domain Generalization and Adaptation. ArXiv:2203.11819 [cs].

Kim, I.; Kim, Y.; and Kim, S. 2020. Learning Loss for Test-Time Augmentation. ArXiv:2010.11422 [cs].

Kobayashi, S. 2018. Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 452–457. New Orleans, Louisiana: Association for Computational Linguistics.

Koh, P. W.; Sagawa, S.; Marklund, H.; Xie, S. M.; Zhang, M.; Balsubramani, A.; Hu, W.; Yasunaga, M.; Phillips, R. L.; Beery, S.; Leskovec, J.; Kundaje, A.; Pierson, E.; Levine, S.; Finn, C.; and Liang, P. 2020. WILDS: A Benchmark of in-the-Wild Distribution Shifts. *CoRR*, abs/2012.07421.

Kumar, V.; Choudhary, A.; and Cho, E. 2021. Data Augmentation using Pre-trained Transformer Models. ArXiv:2003.02245 [cs].

Lazaridou, A.; Kuncoro, A.; Gribovskaya, E.; Agrawal, D.; Liska, A.; Terzi, T.; Gimenez, M.; d'Autume, C. d. M.; Kocisky, T.; Ruder, S.; Yogatama, D.; Cao, K.; Young, S.; and Blunsom, P. 2021. Mind the Gap: Assessing Temporal Generalization in Neural Language Models. ArXiv:2102.01951 [cs].

Lu, H.; Shanmugam, D.; Suresh, H.; and Guttag, J. 2022. Improved Text Classification via Test-Time Augmentation. *arXiv preprint arXiv:2206.13607*.

Ma, E. 2019. NLP Augmentation. https://github.com/makcedward/nlpaug. Accessed: 2022-10-25.

Ma, X.; Xu, P.; Wang, Z.; Nallapati, R.; and Xiang, B. 2019. Domain Adaptation with BERT-based Domain Classification and Data Selection. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, 76–83. Hong Kong, China: Association for Computational Linguistics.

Machireddy, A.; Krishnan, R.; Ahuja, N.; and Tickoo, O. 2022. Continual Active Adaptation to Evolving Distributional Shifts. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 3443–3449. ISSN: 2160-7516.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Mishra, S.; Saenko, K.; and Saligrama, V. 2021. Surprisingly Simple Semi-Supervised Domain Adaptation with Pretraining and Consistency.

Molchanov, D.; Lyzhov, A.; Molchanova, Y.; Ashukha, A.; and Vetrov, D. 2020. Greedy Policy Search: A Simple Baseline for Learnable Test-Time Augmentation. ArXiv:2002.09103 [cs, stat].

Pfülb, B. 2022. Continual Learning with Deep Learning Methods in an Application-Oriented Context.

Pérez-Carrasco, M. I.; Protopapas, P.; and Cabrera-Vives, G. 2021. Con^{2}DA: Simplifying Semi-supervised Domain Adaptation by Learning Consistent and Contrastive Feature Representations.

Quinonero-Candela, J.; Sugiyama, M.; Schwaighofer, A.; and Lawrence, N. D. 2008. *Dataset shift in machine learning*. Mit Press.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Shanmugam, D.; Blalock, D.; Balakrishnan, G.; and Guttag, J. 2021. Better Aggregation in Test-Time Augmentation. ArXiv:2011.11156 [cs].

Sohn, K.; Berthelot, D.; Li, C.-L.; Zhang, Z.; Carlini, N.; Cubuk, E. D.; Kurakin, A.; Zhang, H.; and Raffel, C. 2020. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. ArXiv:2001.07685 [cs, stat].

Sugiyama, M.; and Kawanabe, M. 2012. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press.

Sun, Y.; Wang, X.; Liu, Z.; Miller, J.; Efros, A. A.; and Hardt, M. 2020. Test-Time Training with Self-Supervision for Generalization under Distribution Shifts. ArXiv:1909.13231 [cs, stat].

Thakur, N.; Reimers, N.; Daxenberger, J.; and Gurevych, I. 2021. Augmented SBERT: Data Augmentation Method for Improving Bi-Encoders for Pairwise Sentence Scoring Tasks. ArXiv:2010.08240 [cs].

Wang, W. Y.; and Yang, D. 2015. That's So Annoying!!!: A Lexical and Frame-Semantic Embedding Based Data Augmentation Approach to Automatic Categorization of Annoying Behaviors using #petpeeve Tweets. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2557–2563. Lisbon, Portugal: Association for Computational Linguistics.

Wiles, O.; Gowal, S.; Stimberg, F.; Rebuffi, S.-A.; Ktena, I.; Dvijotham, K. D.; and Cemgil, A. T. 2022. A Fine-Grained Analysis on Distribution Shift. In *International Conference on Learning Representations*.

Wu, Q.; Yue, X.; and Sangiovanni-Vincentelli, A. 2021. Domain-agnostic Test-time Adaptation by Prototypical Training with Auxiliary Data.

Yan, H.; Guo, Y.; and Yang, C. 2021. Augmented Self-Labeling for Source-Free Unsupervised Domain Adaptation. 7.

Zhang, M.; Levine, S.; and Finn, C. 2022. MEMO: Test Time Robustness via Adaptation and Augmentation. ArXiv:2110.09506 [cs].

Zhuang, L.; Wayne, L.; Ya, S.; and Jun, Z. 2021. A Robustly Optimized BERT Pre-training Approach with Post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, 1218–1227. Huhhot, China: Chinese Information Processing Society of China.