RethinkCWS: Is Chinese Word Segmentation a Solved Task?

Jinlan Fu†[*] Pengfei Liu♯[*] Qi Zhang†, Xuanjing Huang†
† School of Computer Science, Shanghai Key Laboratory
of Intelligent Information Processing, Fudan University
♯Carnegie Mellon University
{fujl16,qz,xjhuang}@fudan.edu.cn, pliu3@cs.cmu.edu

Abstract

The performance of the Chinese Word Segmentation (CWS) systems has gradually reached a plateau with the rapid development of deep neural networks, especially the successful use of large pre-trained models. In this paper, we take stock of what we have achieved and rethink what's left in the CWS task. Methodologically, we propose a fine-grained evaluation for existing CWS systems, which not only allows us to *diagnose* the strengths and weaknesses of existing models (under the in-dataset setting), but enables us to *quantify* the discrepancy between different criterion and alleviate the negative transfer problem when doing multi-criteria learning. Strategically, despite not aiming to propose a novel model in this paper, our comprehensive experiments on eight models and seven datasets, as well as thorough analysis, could search for some promising direction for future research. We make all codes publicly available and release an interface that can quickly evaluate and diagnose user's models: https://github.com/neulab/InterpretEval.

1 Introduction

Chinese word segmentation (CWS), as a crucial first step in Chinese language processing, has drawn a large body of research (Sproat and Shih, 1990; Xue and Shen, 2003; Huang et al., 2007; Liu et al., 2014). Recent years have seen remarkable success in the use of deep neural networks on CWS (Zhou et al., 2017; Yang et al., 2017; Ma et al., 2018; Yang et al., 2019; Zheng et al., 2013; Chen et al., 2015b,a; Cai and Zhao, 2016; Pei et al., 2014), and the large unsupervised pre-trained models drive the state-of-the-art results to a new level (Huang et al., 2019).

However, the performance of CWS systems gradually reaches a plateau and the development of this field has slowed down. For example, the CWS systems on many existing datasets (e.g. msr, ctb) have achieved $F1$-score higher than 97.0 but with little further improvement. Naturally, a question would be raised: *is CWS a solved task*? When we rethink on what we have achieved so far, we find that there are still some important while rarely discussed unsolved questions for this task:

Q1: Does current excellent performance (e.g. more than 98.0 $F1$-score on the msr dataset) indicate a perfect CWS system, or are there still some limitations? Existing CWS systems are mainly evaluated by a corpus-level metric. The holistic measure fails to provide a fine-grained analysis. As a result, we are not clear about what the strengths and weaknesses of a specific model are.

To address this problem, we shift the traditional trend of holistic evaluation to fine-grained evaluation, in which the notion of the *attribute* (i.e., *word length*) has been introduced to describe a property of each word. Then test words will be partitioned into different buckets, in which we can observe the system's performances under different aspects based on word's attributes (e.g. *long words* will obtain lower $F1$-score).

Q2: Is there a one-size-fits-all system (i.e., best-performing systems on different datasets are the same)? If no, how can we make different choices of model architectures in different datasets? Insights are still missing for how the choices of different datasets influence architecture design.

To answer this question, we make use of our proposed fine-grained evaluation methodology and present two types of diagnostic methods for existing CWS systems, which not only helps us to identify the strengths and weaknesses of current approaches but provides us with more insight about how different choices of datasets influence the model design.

Q3: Now that existing works show CWS sys-

[*] These two authors contributed equally.

Settings	Measures			Application
In-Dataset	Model	Spearman	S^{ρ} (Eq. 2)	Model Diagnosis Sec. 3.6 (Q1,Q2)
		Variance	S^{σ} (Eq. 3)	
	Data	Sys-indep	α^{μ} (Eq. 4)	
		Sys-dep	α^{ρ} (Eq. 5)	
Cross-Dataset	Model	Generali.	$\mathbf{U}$ (Eq. 6)	Multi-Source Transfer Sec. 4.4 (Q3)
	Data	Criterion	Ψ (Eq. 7)	

Table 1: An outline of our paper. *"Generali."*, *"Sys"*, *"in-dep"*, and *"dep"* are the abbreviation for "Generalization", "System", "independent", and "dependent", respectively.

tems can benefit from multi-criteria learning at the cost of negative transfer (Chen et al., 2017; Qiu et al., 2019), can we design a measure to quantify the discrepancies among different criteria and use it to instruct the multi-criteria learning process (i.e., alleviate negative transfer)?

To answer this question, we extend the *in-dataset* evaluation (i.e., a system is trained and tested on the same dataset) to the setting of *cross-dataset*, in which a CWS model trained on one corpus would be evaluated on a range of out-of-domain corpora. On the other hand, it's the above in-dataset analysis (in Q1 & Q2) that helps us to design a measure to quantify the discrepancies of cross-dataset criterion. Empirical results not only show that the measure, calculated solely based on statistics of two datasets, has a higher correlation with cross-dataset performances but also helps us avoid the negative transfer (i.e., selecting the useful parts of source domains as training sets and achieve better results based on fewer training samples)

Our contributes can be summarized as follows: 1) Instead of using a holistic metric, we proposed an attribute-aided evaluation methodology for CWS systems. This allows us to diagnose the weakness of existing CWS systems (e.g., BERT-based models are not impeccable and limited in dealing with words with high label inconsistency). 2) We show that best-performing systems on different datasets are diverse. Based on some proposed quantified measures, we can make good choices of model architectures in different datasets. 3) We quantify the criterion discrepancy between different datasets, which can alleviate the negative transfer problem when performing multi-criteria learning for CWS.

2 Preliminaries

2.1 Task Description

Chinese word segmentation (CWS) was usually conceptualized as a character-based sequence labeling problem. Formally, let $X = \{x_1, x_2, \ldots, x_T\}$ be a sequence of characters, and $Y = \{y_1, y_2, \ldots, y_T\}$ be the output tags. The goal of the task is to estimate the conditional probability: $P(Y|X) = P(y_t|X, y_1, \cdots, y_{t-1})$. Here, y_t usually takes one value of $\{B, M, E, S\}$.

2.2 Attribute-aided Evaluation Methodology

The standard metric of CWS is becoming hard to distinguish the state-of-the-art word segmentation systems (Qian et al., 2016). Instead of evaluating CWS systems based on a holistic metric (F1 score), in this paper, we take a step towards the fine-grained evaluation of the current CWS systems by proposing an attribute-aided evaluation method. Specifically, we first introduce the notion of *attributes* to characterize the properties of the test words. Then, the test set will be divided into different subsets, and the overall performance could be broken down into several interpretable *buckets*. Below, we will introduce the *seven* attributes that we have explored to depict the word in diverse aspects. Fig. 1 gives an example for the test word "图书馆".

Aspect-I: Intrinsic nature We can characterize a word based on its (or the sentence it belongs to) constitute features. Here, we define three attributes: word length (`wLen`); sentence length (`sLen`); OOV density (`oDen`): the number of words outside the training set in a sentence divided by sentence length.

Aspect-II: Familiarity We introduce a notion of familiarity to quantify the degree to which a test word (or its constituents) has been seen in the training set. Specifically, the familiarity of a word can be calculated based on its *frequency* in the training set. For example, in Fig. 1, if the frequency in the training set of the test word 图书馆 (library) is `0.3`, the attribute of word frequency of 图书馆 will be `0.3`. In this paper, we consider two kinds of familiarity: word frequency (`wFre`); character frequency (`cFre`).

Aspect-III: Label consistency In this paper, we attempt to design a measure that can quantify the degree of label consistency phenomenon (Fu et al., 2020; Gong et al., 2017; Luo and Yang, 2016; Chen et al., 2017) for each test word (or character). Here, we investigate two attributes for label consistency: label consistency of word (`wCon`); label consistency of character (`cCon`). Following,

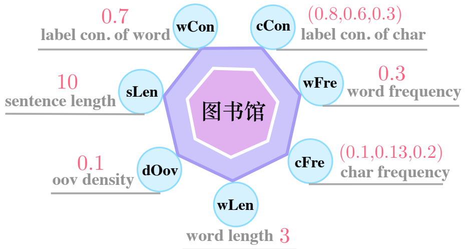


Figure 1: The attribute definition of the word “图书馆(library)” in the sentence: “图书馆在节假日会关闭(The library is closed on holidays”, and its ground truth label is BME. The text in the circle is the abbreviation of the attribute name, and the text in gray and in pink is the full name and the attribute value, respectively. *con.* in grey denotes *consistency*.

we give the definition of label consistency of word, and the label consistency of character can be defined in a similar way. Specifically, we refer to w_i^k as a test word with label k, whose label consistency $\psi(w_i^k)$ is defined as:

$$\psi(w_i^k, D^{tr}) = \begin{cases} 0 & |w_i^{tr}| = 0 \\ \frac{|w_i^{tr,k}|}{|w_i^{tr}|} & \text{otherwise} \end{cases} \tag{1}$$

where $|w_i^{tr,k}|$ represents the occurrence of word w_i with label k in the training set, and D^{tr} is the training set. For example, in Fig. 1, in the training set, “图书馆 (library)” is labeled as BME 7 times, and BMM 3 times, so ψ (“ 图书馆^{BME}”) $= 7/10 = 0.7$, and ψ (“图书馆^{BMM}”) $= 3/10 = 0.3$.

3 Investigation on In-dataset Setting

3.1 Setup

This section focuses on the *in-dataset* setting, in which each CWS model will be trained and test on the same dataset.

Datasets We choose seven mainstream datasets from SIGHAN2005 [1] and SIGHAN2008 [2], in which cityu and ckip are traditional Chinese, while msr, pku, ctb, ncc and sxu are simplified Chinese. We map traditional Chinese characters to simplified Chinese in our experiment. The details of the seven datasets used in this study are described in Chen et al. (2017).

Models We choose typical instances as analytical objects, which vary in terms of the following aspects: 1) character encoders: ELMo (Peters et al., 2018), BERT (Devlin et al., 2018); 2) bigram encoder: Word2Vec (Mikolov et al., 2013), averaging the embedding of two contiguous characters; 3) sentence encoders: LSTM (Hochreiter and Schmidhuber, 1997), CNN (Kalchbrenner et al., 2014); 4) decoders: MLP, CRF (Lample et al., 2016; Collobert et al., 2011). The name of combination of models in in a detailed setting in Tab.2.

3.2 Measures

Here, we refer to $M = \{m_1, \cdots, m_{N_m}\}$ as a set of **models** and $P = \{p_1, \cdots, p_{N_p}\}$ as a set of **attributes**. As described above, the test set could be split into different **buckets** $B = \{B_1^j, \cdots, B_{N_b}^j\}$ based on an attribute p_j. We introduce a performance table $\mathbf{V} \in \mathbb{R}^{N_m \times N_p \times N_b}$, in which $\mathbf{V}_{ijk}$ represents the performance of i-th model on the k-th sub-test set (bucket) generated by j-th attribute.

Model-wise The model-wise measure aims to investigate whether and how the attributes influence the performance of models with different choices of neural components. Formally, we characterize how the j-th attribute influences the i-th model based on two statistical variables: Spearman's rank correlation coefficient Spear (Mukaka, 2012) and standard deviation Std, which can be defined as:

$$\mathbf{S}_{i,j}^{\rho} = \mathrm{Spear}(\mathbf{V}[i, j:], R_j), \tag{2}$$

$$\mathbf{S}_{i,j}^{\sigma} = \mathrm{Std}(\mathbf{V}[i, j:]), \tag{3}$$

where R_j is the rank values of buckets based on j-th attribute. Intuitively, $\mathbf{S}_{i,j}^{\rho}$ reflects the degree to which the i-th model positively (negatively) correlates with j-th attribute while $\mathbf{S}_{i,j}^{\sigma}$ indicates the degree to which this attribute influences the model.

Dataset-wise The dataset-wise measures aim to characterize a dataset with different attributes quantitatively. We utilize two types of measures to build the connection between datasets and attributes: system-independent measure α^{μ}, and system-dependent measures α^{ρ} and α^{σ}.

1) *system-independent measure* reflects intrinsic statistics of the datasets, such as the average word length of the whole dataset. It can be formally defined as:

$$\alpha_j^{\mu} = \frac{1}{N_w} \sum_{i}^{N_w} \mathrm{Attr}(w_i, j), \tag{4}$$

where N_w is the number of test words and $\mathrm{Attr}(w_i, j)$ is the value of attribute j for word w_i.

2) *system-dependent measures* quantify the degree to which each attribute influences the CWS system on a given dataset. For example, “does the

[1] http://sighan.cs.uchicago.edu/bakeoff2005/

[2] https://www.aclweb.org/mirror/ijcnlp08/sighan6/chinesebakeoff.htm

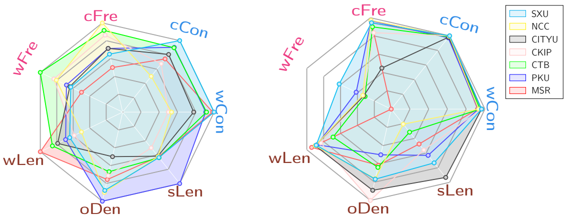
Model	Character				Bigram			SenEnc.		Dec.		Holistic Evaluation (Overall F1)						
	rand	w2v	elmo	bert	none	avg	w2v	lstm	cnn	crf	mlp	msr	pku	ctb	ckip	cityu	ncc	sxu
CrandBavgLstmCrf	√					√		√		√		96.21	**94.22**	**95.32**	**92.81**	93.54	92.01	94.87
Cw2vBavgLstmCrf		√				√		√		√		96.46	94.10	95.08	92.81	93.67	92.04	94.71
Cw2vBavgLstmMlp		√				√		√			√	96.41	92.74	94.09	91.40	93.25	92.00	93.16
Cw2vBavgCnnCrf		√				√			√	√		96.48	93.99	94.72	92.73	**93.72**	**92.64**	94.36
Cw2vBw2vLstmCrf		√					√	√		√		**96.66**	94.19	95.14	92.46	93.70	92.24	**94.97**
CelmBnonLstmMlp			√		√			√			√	96.23	95.33	96.77	94.83	96.44	93.21	96.47
CbertBnonLstmMlp				√	√			√			√	98.19	96.47	**97.68**	**96.23**	97.09	95.77	97.49
CbertBw2vLstmMlp		√		√			√	√			√	**98.20**	96.52	97.65	96.18	97.07	**95.78**	**97.51**
Huang et al. (2019)												97.90	**96.60**	97.60	—	**97.60**	—	97.30

Table 2: Neural CWS systems with different architectures and pre-trained knowledge studied in this paper. We exclude systems based on joint training to make a fair comparison in the in-dataset setting. For the model name, "C" refers to "Character" and "B" refers to "Bigram". Intuitively, the models are named based on their constituents. For example, *Cw2vBw2vLstmCrf* denotes a model's character and the bigram feature is initialized by pre-trained embeddings using Word2Vec, and sentence encoder, as well as the decoder, are LSTM and CRF, respectively. We perform a Friedman test at p = 0.05 on model- (row-) wise and data- (column-)wise. The testing results are $p(\text{model} - \text{wise}) = 2.26 \times 10^{-6} < 0.05$ and $p(\text{data} - \text{wise}) = 8.42 \times 10^{-8}$. Therefore, the results of model-wise and data-wise have passed the significance testing.

attribute word length matter for the CWS system trained on pku dataset?". To achieve this, we design the following measures:

$$\alpha_j^\rho = \frac{1}{N_m} \sum_{i}^{N_m} |\mathbf{S}_{i,j}^\rho|, \tag{5}$$

where N_m is the number of evaluated models. Intuitively, a higher absolute value of $\alpha_j^\rho \in [-1, 1]$ suggests that attribute j is a crucial factor, greatly influencing the performance of CWS systems. For example, if $\alpha_{wLen}^\rho = 0.95$, it means word length is a major factor that influences the CWS performance.



(a) α^μ (b) α^ρ

Figure 2: Illustration of dataset biases characterized by task-independent measure α^μ and task-dependent measures α^ρ. We normalize α^μ on each attribute by divide the maximum α^μ on six datasets, and $\alpha^\rho \in [0, 1]$.

3.3 Analysis of Holistic Evaluation

Before giving a fine-grained analysis, we present the holistic results of different models on different datasets. As shown in Tab. 2, we can observe that **there is no one-size-fits-all model: best-performing systems on different datasets frequently consist of diverse components**. This naturally raises a question: how can pick up appropriate models for different datasets?

3.4 Analysis of Dataset Biases

Before the analysis, we conduct a statistical significance test with the Friedman test (Zimmerman and Zumbo, 1993) at $p = 0.05$, to examine whether the performance of different buckets partitioned by an attribute is significantly different for a given dataset. The results are shown in the Appendix. We find that the performance of different buckets partitioned by an attribute is significantly different ($p < 0.05$), which holds for all the datasets.

1) ***Label consistency* and *word length* have a more consistent impact on CWS performance.** The common parts of the radar charts Fig. 2 (b) illustrate that no matter which datasets are, label consistency attributes (wCon, cCon) and word length (wLen) are highly correlated with CWS performance (higher α^ρ). This suggests that the learning difficulty of CWS systems is commonly influenced by label consistency and word length.

2) ***Frequency* and *sentence length* matters but are minor factors** The outliers in the radar chart (Fig. 2 (b)) show the peculiarities of different corpora. On attributes: sLen, wFre, oDen, the extent to which different datasets are affected varies greatly. For example, the dataset ckip is distinctive with the highest value of α_{oDen}^ρ, which can explain why character pre-training shows no advantage while the CRF layer contributes a lot.

3.5 Analysis of Model Biases

Similar to the above section, we perform the Friedman test at $p = 0.05$. We give detailed significance testing results in the Appendix. Tab. 3 gives an illustration of model biases characterized by measures $\mathbf{S}^{\rho}_{i,j}$ and $\mathbf{S}^{\sigma}_{i,j}$. The values in grey denote the given model on the specific attribute does not pass the significance test ($p \geq 0.05$). Below, we will highlight some observations.

ELMo-based Models can make better use of the context information that long sentences carry. Regarding the attribute of `sLen` (sentence length), two models *CelmBnonLstmMlp* and *CbertBnonLstmMlp* pass the significance test. Additionally, we observe only ELMo (*CelmBnonLstmMlp*) shows a strong positive correlation with sentence length, referring to Tab. 3.

Contextualized models could reduce the negative effect of OOV density and remedy the deficiency of MLP decoder. a) The performances of non-contextualized models (i.e. word2vec) strongly correlate with the `oDen` (density of OOV words) attribute. When equipped with BERT or ELMo, the model still could provide each OOV word with a meaningful representation on the fly based on its context. b) We observe that the model *Cw2vBavgLstmMlp* is strong correlated with `wCon` and `wLen` with highest values of $\mathbf{S}^{\sigma}$ (referring to Tab. 3 with bolded value), suggesting that models with MLP layer are unstable when generalizing to the hard cases (words with lower value of `wCon` and higher value of `wLen`). However, once augmented with contextualized models, systems with MLP decoder also work well.

Model	F1	Spearmanr							Standard Deviation						
		wCon	cCon	cFre	wFre	wLen	oDen	sLen	wCon	cCon	cFre	wFre	wLen	oDen	sLen
CrandBavgLstmCrf	94.14	92	99	88	33	-85	-82	20	13	9.3	2.4	6.4	13	1.6	0.6
Cw2vBavgLstmCrf	94.12	93	99	91	33	-85	-86	18	13	10	2.4	7.3	13	1.8	0.6
Cw2vBavgLstmMlp	93.29	95	98	93	37	-86	-76	8.9	19	11	3.1	7.9	15	2.7	1.2
Cw2vBavgCnnCrf	94.09	96	99	92	35	-86	-73	17	15	9.4	2.5	7.0	14	1.5	0.7
Cw2vBw2vLstmCrf	94.20	93	99	90	33	-89	-85	28	13	10	2.4	7.5	13	1.9	0.6
CelmBnonLstmMlp	95.61	95	98	78	31	-82	-44	73	9.0	5.1	1.4	4.5	8.2	1.5	0.5
CbertBnonLstmMlp	96.99	96	98	74	34	-88	-30	39	6.2	3.7	1.0	2.8	5.8	1.2	0.3
CbertBw2vLstmMlp	97.00	96	99	77	30	-86	-29	37	6.3	3.9	1.0	2.8	5.8	1.2	0.3

Table 3: Illustration of model biases characterized by model-wise measure (Percentage) $\mathbf{S}^{\rho}_{i,j}$ and $\mathbf{S}^{\sigma}_{i,j}$. Here, we average the F1, $\mathbf{S}^{\rho}_{i,j}$ and $\mathbf{S}^{\sigma}_{i,j}$ on seven datasets. The values in gray denotes the given model on the specific attribute does not pass the significance test ($p \geq 0.05$). The values in orange and in blue support observation 1 and observation 2, respectively.

3.6 Application: Model Diagnosis

Model diagnosis is the process of identifying where the model works well and where it worse (Vartak et al., 2018). We present two types of diagnostic methods: ***self-diagnosis*** and ***aided-diagnosis***. *self-diagnosis* aims to locate the bucket on which the input model has obtained the worst performance with respect to a given attribute. For *aided-diagnosis*, supposing that the holistic performance of two models satisfies: $A > B$. Then *Aided-diagnosis*(A,B) will first look for a bucket, on which the performance satisfies: $A < B$. If there is no qualified bucket, then the bucket, on which model A has achieved the best performance, will be returned.

Below, we will give a diagnostic analysis of some typical models shown in Tab. 4. The others are shown in the Appendix.

Self-diagnosis: BERT-based models are not impeccable. The first row in Tab. 4 shows the diagnosis of model *CbertBnonLstmMlp*, in which the x-ticklabel represents the bucket value of a specific attribute (e.g. `wLen`: word length) on which system has achieved worst performance. The blue bins represent the worst performance, while red bins denote the gap between worst and best performance. For example, the first histogram in the first row denotes that *CbertBnonLstmMlp* achieved the worst performance on attribute `wCon` with value `S`.

We observe that there is a huge performance drop on all the datasets when the test samples are with the attribute values: `wCon=S` (low label consistency of words), `cCon=S` (low label consistency of characters), `wLen=L` (long words). This suggests that contextualized information brought from BERT is not insufficient to deal with low label consistency and long words. To address this challenge, more efforts should be made on learning algorithms or data augmentation strategies.

Aided diagnosis: BERT v.s ELMo The second row in Tab. 4 shows the comparing between BERT and ELMo and we observed 1) BERT outperforms ELMo in the bucket of `wCon=S` (low label consistency of words) a lot on all datasets, suggesting that the benefit of BERT mainly comes from the processing of low label consistency of words. 2) When the OOV density of a sentence is high enough, BERT will lose its superiority. As shown in Tab. 4, BERT performs worse than ELMo in the bucket of `oDen=L` on the `pku` dataset whose average OOV density (α^{μ}_{oDen}) is the highest one

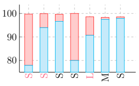
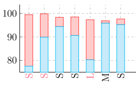
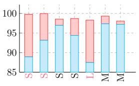
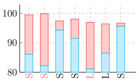
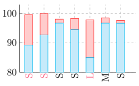
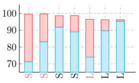
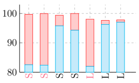
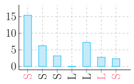
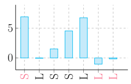
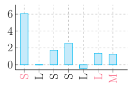
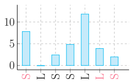
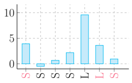
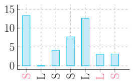
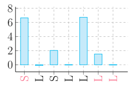
	msr	pku	ctb	ckip	cityu	ncc	sxu
	wCon cCon cFre wFre wLen oDen sLen	wCon cCon cFre wFre wLen oDen sLen	wCon cCon cFre wFre wLen oDen sLen	wCon cCon cFre wFre wLen oDen sLen	wCon cCon cFre wFre wLen oDen sLen	wCon cCon cFre wFre wLen oDen sLen	wCon cCon cFre wFre wLen oDen sLen
Overall F1	A: 98.19	A: 96.47	A: 97.68	A: 96.23	A: 97.09	A: 95.77	A: 97.49
A:CbertBnonLstmMlp Self-diagnosis							
Overall F1	A:98.19; B:96.23	A:96.47; B:95.33	A:97.68; B:96.77	A:96.23; B:94.83	A:97.09; B:96.44	A:95.77; B:93.21	A:97.49; B:96.47
A: *CbertBnonLstmMlp* B: *CelmBnonLstmMlp* Aided-diagnosis							

Table 4: Diagnosis of different CWS systems. For ease of presentation, we attribute values are classified into three categories: **small**(S), **middle**(M), and **large**(L). Regarding *Self-diagnosis*, the x-ticklabel represents the bucket value of a specific attribute (e.g. `wLen`: word length) on which the system has achieved the worst performance. The blue bins represent the worst performance, while red bins denote the gap between worst and best performance. Regarding *Aided-diagnosis*, the bins below the line "$y = 0$" represent the largest gap that model A is less than model B. By contrast, the bins above the line "$y = 0$" denote the largest gap that model A is better than model B. x-ticklabels in red indicate that the corresponding bins will be used for analysis in Sec. 3.6.

(as shown in Fig. 2 (a)). To explain this, we take a closer look at the testing samples in the `pku` with high OOV density: "仰泳100米和400米" (backstroke 100m and 400m), "10月1日， 北京 (October 1, Beijing)" . BERT, as multi-layer Transformers, is challenging to collect sufficient context to understand these cases. 3) BERT is inferior to ELMo in dealing with long sentences. As shown in Tab. 4, BERT obtain lower performance in the bucket of `sLen=L` on `pku` and `sxu` datasets, whose average lengths (α^{μ}_{sLen}) are the highest two.

4 Investigation on Cross-dataset Setting

The above in-dataset analysis aims to interpret model bias and dataset bias based on individual datasets. In many real-world scenarios, we need to transfer a trained model to a new dataset or domain, which requires us to understand the cross-dataset generalization behavior of current systems. In this section, our investigation on cross-dataset generalization is driven by two questions: 1) How different architectures (i.e. *Cw2vBavgLstmCrf*) of CWS systems influence their cross-dataset generalization ability? 2) Now that we have found the common factor (`label consistency`) that affects model performance across different datasets in the previous section, can we design a measure based on it and use it to interpret cross-data generalization? We will detail our exploration below.

4.1 Setup

This section focuses on the *zero-shot* setting: a model with specified architecture trained on one dataset (e.g. `pku`) will be evaluated on a range of other datasets (e.g. `ctb`). To better understand the generalization behavior of CWS systems and the relation between different datasets, we first define several measures to quantify our observations.

4.2 Measures

Similar to Sec.3.2, we refer to N_d as the number of all datasets and N_m as the number of architectures. The cross-dataset performance can be recorded by the following matrix:

$$\mathbf{U} \in \mathbb{R}^{N_d \times N_d \times N_m} \qquad (6)$$

Quantifying System's Cross-dataset Generalization Intuitively, $\mathbf{U}_{ijk} = 0.65$ represents that we have adopted the architecture k (i.e. *Cw2vBavgLstmCrf*) to learn a model on the training set of i (e.g. `pku`), and the performance on test set of j (e.g., `msr`) is 0.65.

We do some simple numerical processing on matrix $\mathbf{U}$ to make the meaning of variables more intuitive: $\hat{\mathbf{U}}_{ijk} = (\mathbf{U}_{jjk} - \mathbf{U}_{ijk})/\mathbf{U}_{jjk}$. $\hat{\mathbf{U}}_{pku,msr,k} = 0.2$ suggests that, both tested on `msr`, the model with architecture k trained on `pku` is relatively lower than that trained on `msr` by 0.2. Usually, a lower value of $\hat{\mathbf{U}}$ is suggestive of better zero-shot generalization ability.

Quantifying Discrepancies of Cross-dataset Criterion To measure the discrepancy of segmentation criteria between any pair of training data D^{tr}_A and test data D^{te}_B, we extend the *label consistency of word* (defined in Sec. 2.2) to corpus-level by computing its expectation on a given training-test dataset pair. Base on Eq. 1, we defined the measure Ψ as:

$$\Psi(D^{tr}_A, D^{te}_B) = \sum_{i \in N_w} \psi(w^{te,k}_i, D^{tr}_A) * \mathrm{freq}(w^{te,k}_i) \qquad (7)$$

in which $\psi(\cdot)$ (defined in Eq. 1) is a function to calculate the label consistency for a test word $w^{te,k}_i$.

N_w denotes the number of unique test words and $\text{freq}(w_i^{te,k})$ is the frequency of the test word.

A lower value of $\Psi(D_A^{tr}, D_B^{te})$ suggests a larger discrepancy between the two datasets. For example, $\Psi(D_{msr}^{tr}, D_{msr}^{te}) = 78.0$ and $\Psi(D_{msr}^{tr}, D_{pku}^{te}) = 75.5$, indicating that the discrepancy between msr's training set and msr's test set is smaller than the discrepancy between msr's training set and pku's test set.

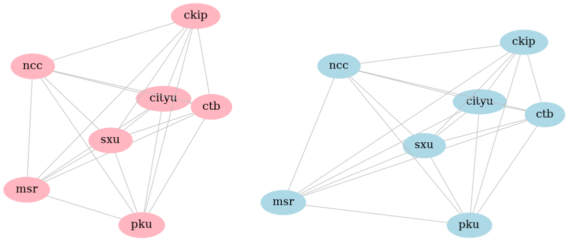
Data	Data-wise (Ψ)								Model-wise ($\hat{U}$)							
	msr	**pku**	**ctb**	**ckip**	**cityu**	**ncc**	**sxu**	avg	**msr**	**pku**	**ctb**	**ckip**	**cityu**	**ncc**	**sxu**	avg
msr	**78.0**	69.8	67.6	64.4	72.9	67.8	71.2	70.2	0	9.8	14	13	7.5	5.8	9.5	8.4
pku	75.5	**77.3**	71.9	66.2	75.6	68.1	74.5	72.8	11	0	7.5	8.3	3.8	7.6	5.3	6.2
ctb	72.5	71.7	**77.4**	71.0	**76.4**	67.5	74.7	73.0	14	8.1	0	4.1	2.1	10	5.8	6.4
ckip	69.2	67.7	73.1	**74.1**	73.3	67.0	71.2	70.8	16	10	4.4	0	4.0	9.9	7.7	7.4
cityu	70.2	67.8	73.3	70.0	**76.3**	65.9	72.3	70.8	14	10	5.2	5.1	0	9.2	6.2	7.1
ncc	74.2	70.5	70.0	68.2	73.6	**74.3**	73.4	72.0	11	11	12	10	7.8	0	7.7	8.5
sxu	72.6	72.1	71.4	66.9	75.5	69.1	**78.1**	72.2	13	7.4	7.5	8.1	3.0	8.1	0	6.8
avg	73.2	71.0	72.1	68.7	74.8	68.5	73.6	71.7	11	8.1	7.2	7.0	4.0	7.3	6.0	7.3

Table 5: The relationship between different pairs of datasets measured by data-wise Ψ and model-wise $\hat{U}_k$. Here k represents the model *Cw2vBavgLstmCrf*.

4.3 Analysis

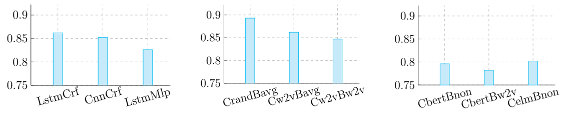
Tab. 5 illustrate the relationship between different train-test pair using data-wise Ψ and model-wise $\hat{U}_k$. To test whether the expectation of label consistency is a factor that can be used to characterize cross-dataset generalization, we perform a Friedman test at $p = 0.05$. Each group of samples for significance testing is obtained by changing the test-set for a given train-set (we have 7 groups of testing samples corresponding to the 7 columns data of Ψ in Tab. 5). The testing result is $p = 0.011 < 0.05$, therefore, Ψ can be utilized to describe the feature of a cross-dataset pair.

The distance between different datasets can be quantitatively characterized by Ψ. 1) As shown in Tab. 5, nearly all highest values are achieved on the diagonal except the row of cityu. Ψ(cityu, cityu) is slightly lower than Ψ(ctb, cityu), indicating the training sets of ctb and cityu are quite close. As shown in Fig. 3(a), we do find cityu is closet to ctb. 2) ctb achieves the highest value in the "avg"-column of Tab. 5 in red, which shows taking ctb as the source domain, the average distance to test sets of other corpora is the smallest. Similarly, if cityu is regarded as the target domain, then the average distance from other training sets to it is the smallest. 3) As shown in Fig. 3(a), sxu, cityu, and ctb cluster together, surrounded by other datasets ckip, ncc, and pku remotely, suggesting that these neighbor datasets have the similar distribution.



(a) Data-wise (Ψ) (b) Model-wise (U)

Figure 3: 2D-visualization of the distances between datasets computed based on data-wise measure Ψ and model-wise U averaging on seven datasets, respectively. The weight of between dataset i and j is transformed into an undirected edge based on: $\frac{Z_{ij}}{Z_{jj}} + \frac{Z_{ji}}{Z_{ii}}$ and Z can be Ψ and U, in which the distance computed based on U is the average on eight models.



(a) Cw2vBavg (b) LstmCrf (c) LstmMlp

Figure 4: The Spearman's rank correlation coefficient between Ψ and the U_k.

The measure Ψ could be used to interpret the domain shift. As shown in Tab. 5, we find the value of Ψ could reflect the changing trends of $\hat{U}$. Similarly, as shown in Fig.3, impressively, these two graphs obtained in totally different ways are so close: Fig.3 (a) is computed purely based on intrinsic statistics of the dataset, while Fig.3 (b) is obtained based on model outputs. These qualitative results show our proposed measure Ψ could be used to explain the discrepancies across datasets.

To get a more convincing observation, we additionally conduct a quantitative analysis. Specifically, we calculate the Spearman's rank correlation coefficient between Ψ and the U_k. The results all shown in Fig. 4 (a-c). Encouragingly, we find that no matter which CWS system, the cross-dataset performances of them are highly correlated with our proposed measure of Ψ.

4.4 Application: Multi-source Transfer

Given a target domain D_t, the above quantitative and qualitative analysis shows that the measure Ψ can be used to quantify the importance of different source domains $D_{s_1}, \cdots, D_{s_N}$, therefore allowing us to select suitable ones for data augmentation.

Next, we will show how to use the Ψ to make

Algorithm 1 Decoding Process for Dataset Order

```
Require: Target domain $D_t = \{D_t^{tr}, D_t^{dev}\}$; a sequence
    of source domains $\{D_{s_1}, D_{s_2}, \dots, D_{s_N}\}$; indexes of
    source domains $K = \{1 \cdots N\}$;measure $\Phi$
Require: $\hat{K} \leftarrow \{\}$; $\hat{D} \leftarrow D_t^{tr}$
for $k \in K$ do
    if Max-select then
        $\hat{k} = \mathrm{argmax}_{k \in K \wedge k \notin \hat{K}} \Phi(\hat{D} + D_{s_k}, D_t^{dev})$
    else if Min-select then
        $\hat{k} = \mathrm{argmin}_{k \in K \wedge k \notin \hat{K}} \Phi(\hat{D} + D_{s_k}, D_t^{dev})$
    else if Rand-select then
        $\hat{k} = \mathrm{Random}_{k \in K \wedge k \notin \hat{K}} \Phi(\hat{D} + D_{s_k}, D_t^{dev})$
    end if
    $\hat{K} = \hat{K} + \{\hat{k}\}$                                ▷ EnQueue
    $\hat{D} = \hat{D} + D_{s_{\hat{k}}}$
end for
    return $\hat{K}$
```

better choices of source domains from the other candidates. We take ctb as the tested object and continuously increase the training samples of above the seven datasets in three different ways: *Rand-*, *Max-*, and *Min-select*. Alg. 1 shows the decoding process for the dataset order. We choose the multi-criteria segmenter proposed by Chen et al. (2017) as our training framework for multiple datasets.

Result Fig. 5 illustrates the changes in F1-score as more source domains are introduced in three different orders. We do a Friedman test with the null hypothesis that the order of training set introduced had no influence on the performance of a given model. The significance testing result shows that the training set introduced with Max-, Min-, and Rand-select are significantly different ($p = 8.0 \times 10^{-3} < 0.05$). We can observe from Fig. 5 that: *More training samples are not a guarantee of better results for CWS models due to the criteria discrepancy between different datasets.*

Specifically, the *Max-select* operation helps us find an optimal set of source domains (ctb, sxu, ncc, cityu), on which the model could achieve the best results, outperforming Chen et al. (2017)'s result by a significant margin, which trained on nine datasets (two more than ours). Regarding the two baseline decoding strategies (*Min-select* and *Rand-select*), we find the best performance on ctb are both obtained when all seven training sets are used. The above observations indicate that, when we introduce multiple training sets for data augmentation, the order of the distance between training and development sets can help us select which parts of source domains are useful. And Ψ, we proposed in this paper, is an effective measure to quantify this order (without learning process), providing a novel solution for multi-source transfer learning.

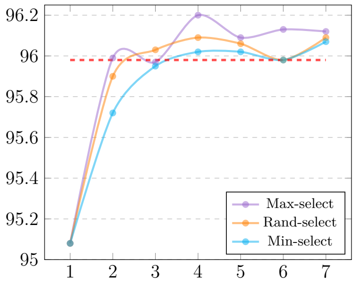


Figure 5: The changing of F1-score as more source domains are introduced in three different orders: *Max-*, *Min-*, and *Rand-select*. The red dotted line is the result reported by Chen et al. (2017) with the same model, trained on nine datasets.[1]

5 Discussion

We summarize the main observations from our experiments and try to give preliminary answers to our proposed research questions:

Does existing excellent performance imply a perfect CWS system? No. Beyond giving this unsurprising answer, we present an interpretable evaluation method to help us diagnose the weaknesses of existing top-performing systems and relative merits between two systems. For example, we find even top-scoring BERT-based models still cannot deal with the words with low label consistency or long words well, and BERT is inferior to ELMo as an encoder in dealing with long sentences.

Is there a one-size-fits-all system? No (Best-performing systems on different datasets frequently involve diverse neural architectures). Although this question can be answered relatively easily by simply looking at the overall results of different systems in diverse data sets (Sec.2), we take a step further to how to make choices of them (BERT v.s ELMo, LSTM v.s CNN)) by conducting dataset bias-aware *Aided-diagnosis* (Sec.3.6).

Can we design a measure to quantify the discrepancies among different criteria? Yes. We first verify that the *label consistency* of words and *word length* have a more consistent impact on CWS performance. Based on this, we design a measure to quantify the distance between different datasets, which correlates well with the cross-dataset performance and can be used for multi-source transfer learning, help us avoid the negative transfer.

[1] To make a fair comparison, all results are implemented based on their public code.

6 Acknowledgements

The authors wish to thank Zhenghua Li and Meishan Zhang for their helpful comments and carefully proofreading of our manuscript. This work was partially funded by China National Key R&D Program (No. 2017YFB1002104，2018YFC0831105, 2018YFB1005104), National Natural Science Foundation of China (No. 61976056, 61751201, 61532011), Shanghai Municipal Science and Technology Major Project (No.2018SHZDZX01), Science and Technology Commission of Shanghai Municipality Grant (No.18DZ1201000, 17JC1420200).

References

Deng Cai and Hai Zhao. 2016. Neural word segmentation learning for chinese. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 409–420.

Xinchi Chen, Xipeng Qiu, Chenxi Zhu, and Xuanjing Huang. 2015a. Gated recursive neural network for chinese word segmentation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1744–1753.

Xinchi Chen, Xipeng Qiu, Chenxi Zhu, Pengfei Liu, and Xuanjing Huang. 2015b. Long short-term memory neural networks for chinese word segmentation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1197–1206.

Xinchi Chen, Zhan Shi, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-criteria learning for chinese word segmentation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1193–1203.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jinlan Fu, Pengfei Liu, Qi Zhang, and Xuanjing Huang. 2020. Rethinking generalization of neural models: A named entity recognition case study. In *AAAI*, pages 7732–7739.

Chen Gong, Zhenghua Li, Min Zhang, and Xinzhou Jiang. 2017. Multi-grained chinese word segmentation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 692–703.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Chu-Ren Huang, Petr Šimon, Shu-Kai Hsieh, and Laurent Prévot. 2007. Rethinking chinese word segmentation: tokenization, character classification, or wordbreak identification. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 69–72. Association for Computational Linguistics.

Weipeng Huang, Xingyi Cheng, Kunlong Chen, Taifeng Wang, and Wei Chu. 2019. Toward fast and accurate neural chinese word segmentation with multi-criteria learning. *arXiv preprint arXiv:1903.04190*.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of ACL*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT*, pages 260–270.

Yijia Liu, Yue Zhang, Wanxiang Che, Ting Liu, and Fan Wu. 2014. Domain adaptation for crf-based chinese word segmentation using free annotations. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 864–874.

Wencan Luo and Fan Yang. 2016. An empirical study of automatic chinese word segmentation for spoken language understanding and named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 238–248.

Ji Ma, Kuzman Ganchev, and David Weiss. 2018. State-of-the-art chinese word segmentation with bi-lstms. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4902–4908.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mavuto M Mukaka. 2012. A guide to appropriate use of correlation coefficient in medical research. *Malawi Medical Journal*, 24(3):69–71.

Wenzhe Pei, Tao Ge, and Baobao Chang. 2014. Max-margin tensor neural network for chinese word segmentation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 293–303.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 2227–2237.

Peng Qian, Xipeng Qiu, and Xuanjing Huang. 2016. A new psychometric-inspired evaluation metric for chinese word segmentation. 1:2185–2194.

Xipeng Qiu, Hengzhi Pei, Hang Yan, and Xuanjing Huang. 2019. Multi-criteria chinese word segmentation with transformer. *arXiv: Computation and Language*.

Richard Sproat and Chilin Shih. 1990. A statistical method for finding word boundaries in chinese text. *Computer Processing of Chinese and Oriental Languages*, 4(4):336–351.

Manasi Vartak, Joana M F da Trindade, Samuel Madden, and Matei Zaharia. 2018. Mistique: A system to store and query model intermediates for model diagnosis. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1285–1300. ACM.

Nianwen Xue and Libin Shen. 2003. Chinese word segmentation as lmr tagging. In *Proceedings of the second SIGHAN workshop on Chinese language processing-Volume 17*, pages 176–179. Association for Computational Linguistics.

Jie Yang, Yue Zhang, and Fei Dong. 2017. Neural word segmentation with rich pretraining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 839–849.

Jie Yang, Yue Zhang, and Shuailong Liang. 2019. Subword encoding in lattice lstm for chinese word segmentation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2720–2725.

Xiaoqing Zheng, Hanyang Chen, and Tianyu Xu. 2013. Deep learning for chinese word segmentation and pos tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 647–657.

Hao Zhou, Zhenting Yu, Yue Zhang, Shujian Huang, Xin-Yu Dai, and Jiajun Chen. 2017. Word-context character embeddings for chinese word segmentation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 760–766.

Donald W Zimmerman and Bruno D Zumbo. 1993. Relative power of the wilcoxon test, the friedman test, and repeated-measures anova on ranks. *The Journal of Experimental Education*, 62(1):75–86.

A Significance Testing

To conduct the fine-grained evaluation, we divide the words (characters) of the test set into several subsets, which are named buckets in this paper. We perform Friedman significance testing at $p = 0.05$ in dataset-dimension and model-dimension, and the results are shown in Tab. 6 and Tab. 7, respectively. For dataset-dimension (model-dimension), the null hypothesis is that the performance of buckets concerning an attribute has the same means for a given dataset (model).

B Application: Model Diagnosis

Model diagnosis is the process of identifying where the model works well and where it worse. Tab. 8 shows several model diagnoses of different CWS systems. Below, we will give several comparative-diagnostic analysis on some typical models.

LSTM v.s. CNN For the choice of CNN or LSTM, the main factors are sLen (sentence length) and cCon (label consistency of characters), referring to third row of Tab. 8. Besides shorter sentences, we're surprised to find that the *CNN encoder is better at handling ambiguous characters than LSTM*. Generally, we believe that LSTM could provide more long-term information, therefore, achieving disambiguation. However, the above results show that local information is more important to learn these highly ambiguous characters (such as “的”,“了”, “什”) for the CWS task. Based on this, we could explain why CNN outperforms LSTM on ncc (lowest value of α^{u}_{wCon}) while is significantly worse than LSTM on ctb and sxu (large values of α^{u}_{cAmb}).

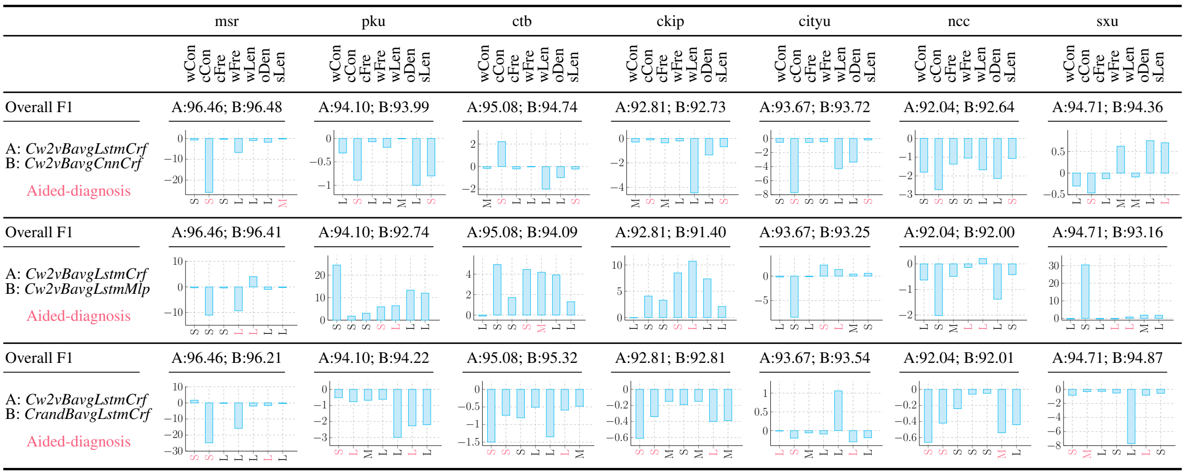
CRF v.s. MLP CRF decoder has no advantage in dealing with unambiguous words compared with MLP, but is superior in processing long (wLen=L) and ambiguous (wCon=S) words, as observed in the fourth row of Tab. 8. Particularly, *Cw2vBavgLstmCrf* outperforms *Cw2vBavgLstmMlp* models by a large margin in the bucket of (wCon=S) on the pku dataset. Based on this, we could explain the difference in

datas	wCon	cCon	cFre	wFre	wLen	oDen	sLen
msr	1.2×10^{-11}	1.0×10^{-11}	2.5×10^{-11}	9.8×10^{-11}	2.0×10^{-10}	5.8×10^{-09}	9.1×10^{-09}
pku	7.2×10^{-12}	1.1×10^{-11}	2.7×10^{-11}	1.5×10^{-10}	8.4×10^{-11}	1.4×10^{-07}	4.8×10^{-08}
ctb	7.2×10^{-12}	8.0×10^{-12}	3.9×10^{-11}	3.6×10^{-10}	1.5×10^{-10}	1.3×10^{-07}	4.9×10^{-07}
ckip	7.2×10^{-12}	7.3×10^{-12}	9.5×10^{-10}	1.0×10^{-10}	8.3×10^{-09}	2.9×10^{-11}	5.5×10^{-05}
cityu	6.2×10^{-12}	1.0×10^{-11}	4.1×10^{-11}	9.6×10^{-11}	4.5×10^{-10}	8.1×10^{-11}	2.3×10^{-10}
ncc	7.2×10^{-12}	7.4×10^{-12}	7.8×10^{-12}	1.6×10^{-10}	2.6×10^{-11}	2.2×10^{-10}	1.7×10^{-09}
sxu	6.3×10^{-12}	9.3×10^{-12}	2.1×10^{-11}	1.3×10^{-10}	7.9×10^{-09}	2.6×10^{-08}	5.5×10^{-08}

Table 6: p-values from the Friedman test. The null hypothesis is that the performance of different buckets with respect to an attribute has the same means for a given **dataset**.

models	wCon	cCon	cFre	wFre	wLen	oDen	sLen
CrandBavgLstmCrf	6.5×10^{-10}	5.4×10^{-10}	6.9×10^{-7}	9.5×10^{-5}	4.5×10^{-5}	1.8×10^{-5}	2.1×10^{-1}
Cw2vBavgLstmCrf	6.5×10^{-10}	6.6×10^{-10}	5.7×10^{-7}	7.8×10^{-5}	2.9×10^{-5}	2.8×10^{-6}	2.0×10^{-1}
Cw2vBavgLstmMlp	6.0×10^{-10}	7.5×10^{-10}	1.4×10^{-7}	1.6×10^{-4}	3.8×10^{-5}	3.3×10^{-4}	4.5×10^{-1}
Cw2vBavgCnnCrf	5.7×10^{-10}	5.1×10^{-10}	4.5×10^{-7}	5.9×10^{-5}	1.1×10^{-4}	1.1×10^{-4}	8.3×10^{-1}
Cw2vBw2vLstmCrf	6.5×10^{-10}	5.2×10^{-10}	3.0×10^{-7}	1.2×10^{-4}	2.3×10^{-5}	6.8×10^{-6}	2.6×10^{-1}
CelmBnonLstmMlp	6.6×10^{-10}	6.5×10^{-10}	1.0×10^{-5}	1.9×10^{-4}	4.5×10^{-4}	4.1×10^{-4}	2.1×10^{-4}
CbertBnonLstmMlp	7.5×10^{-10}	1.4×10^{-9}	3.8×10^{-5}	1.3×10^{-4}	4.4×10^{-5}	1.1×10^{-4}	2.7×10^{-2}
CbertBw2vLstmMlp	6.6×10^{-10}	7.8×10^{-10}	1.5×10^{-5}	1.1×10^{-4}	5.4×10^{-5}	8.0×10^{-3}	6.5×10^{-2}

Table 7: p-values from the Friedman test. The null hypothesis is that the performance of different buckets with respect to an attribute has the same means for a given **model**. The values in pink indicate that the value is greater than $p = 0.05$.



	msr	pku	ctb	ckip	cityu	ncc	sxu
	wCon cCon cFre wFre wLen oDen sLen	wCon cCon cFre wFre wLen oDen sLen	wCon cCon cFre wFre wLen oDen sLen	wCon cCon cFre wFre wLen oDen sLen	wCon cCon cFre wFre wLen oDen sLen	wCon cCon cFre wFre wLen oDen sLen	wCon cCon cFre wFre wLen oDen sLen
Overall F1	A:96.46; B:96.48	A:94.10; B:93.99	A:95.08; B:94.74	A:92.81; B:92.73	A:93.67; B:93.72	A:92.04; B:92.64	A:94.71; B:94.36
A: *Cw2vBavgLstmCrf* B: *Cw2vBavgCnnCrf* Aided-diagnosis							
Overall F1	A:96.46; B:96.41	A:94.10; B:92.74	A:95.08; B:94.09	A:92.81; B:91.40	A:93.67; B:93.25	A:92.04; B:92.00	A:94.71; B:93.16
A: *Cw2vBavgLstmCrf* B: *Cw2vBavgLstmMlp* Aided-diagnosis							
Overall F1	A:96.46; B:96.21	A:94.10; B:94.22	A:95.08; B:95.32	A:92.81; B:92.81	A:93.67; B:93.54	A:92.04; B:92.01	A:94.71; B:94.87
A: *Cw2vBavgLstmCrf* B: *CrandBavgLstmCrf* Aided-diagnosis							

Table 8: Diagnosis of different CWS systems. For ease of presentation, we attribute values are classified into three categories: **small**(S), **middle**(M), and **large**(L). Regarding *Aided-diagnosis*, the bins below the line "$x = 0$" represent the largest gap that model A is less than model B. By contrast, the bins above the line "$x = 0$" denote the largest gap that model A is better than model B. x-ticklabels in red indicate that the corresponding bins will be used for analysis in this section

the holistic F1 results between the above two models.

Cw2vBavg v.s. CrandBavg As shown in the last row of the Tab. 8, we find that pre-trained knowledge does not always help to improve the performance, especially when: 1) the characters or words are highly ambiguous; 2) the OOV density of a sentence is high. Above evidences will help us to explain why *CrandBavg* could achieve better performance measured on the holistic F1 on `ctb`, `sxu` and `pku`. They share a property of much higher value of α^{μ}_{wCon}, α^{μ}_{cCon}, α^{μ}_{oDen} as observed in the Fig. 2 (a).