Zoom Out and Observe: News Environment Perception for Fake News Detection

Qiang Sheng, Juan Cao*, Xueyao Zhang, Rundong Li, Danding Wang, Yongchun Zhu
Key Lab of Intelligent Information Processing of Chinese Academy of Sciences,
Institute of Computing Technology, Chinese Academy of Sciences
University of Chinese Academy of Sciences
{shengqiang18z,caojuan,zhangxueyao19s}@ict.ac.cn
{lirundong20s,wangdanding,zhuyongchun18s}@ict.ac.cn

ACL2022

https://github.com/ICTMCG/News-Environment-Perception/

2022. 4. 24

Reported by Xiaoke Li

Introduction

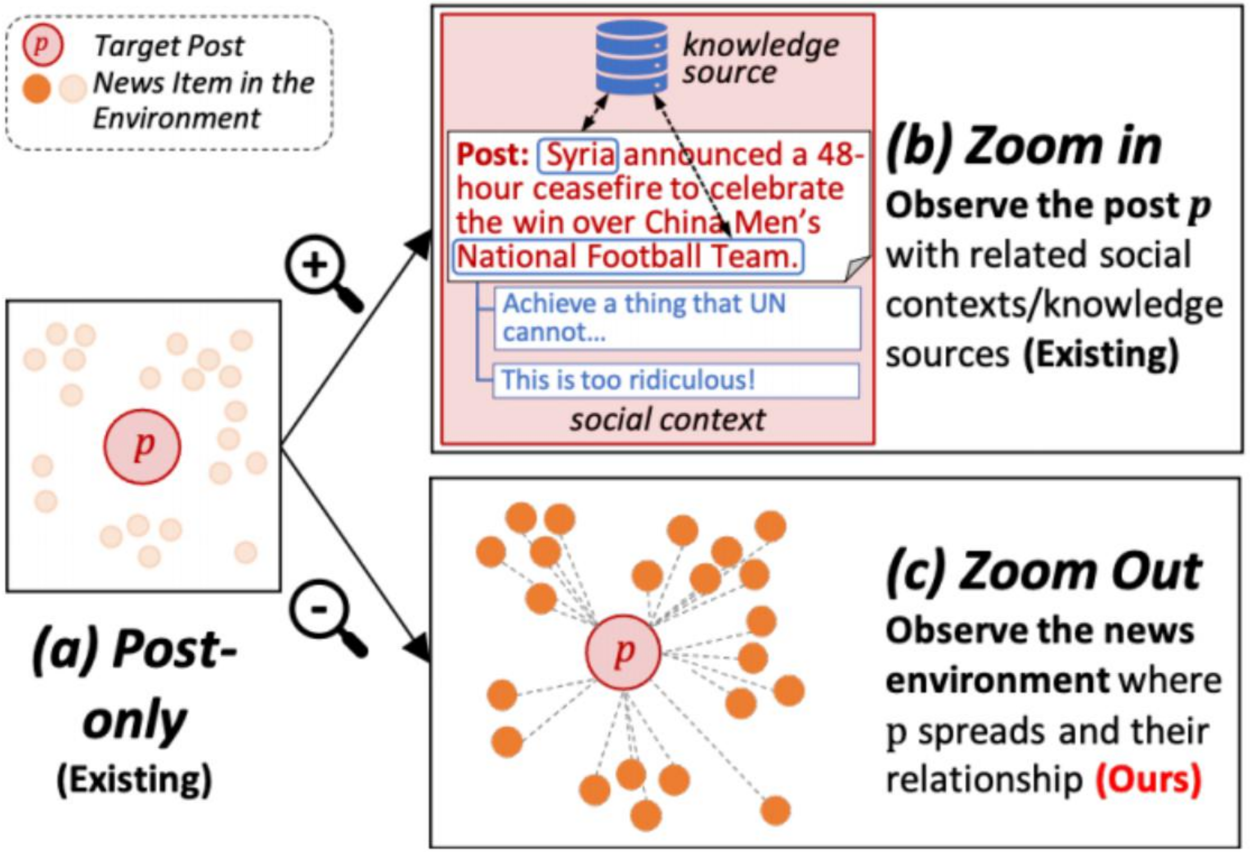
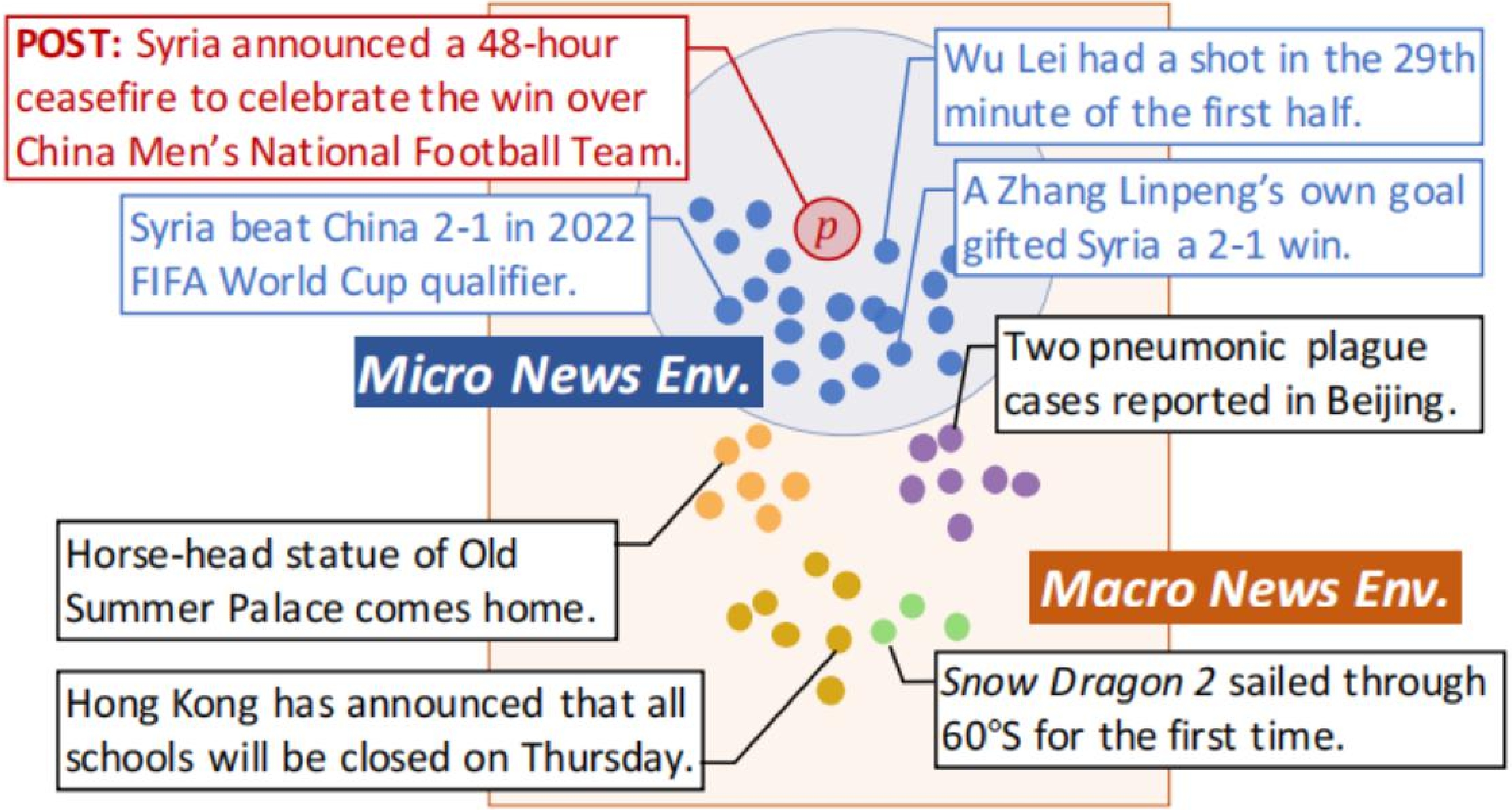
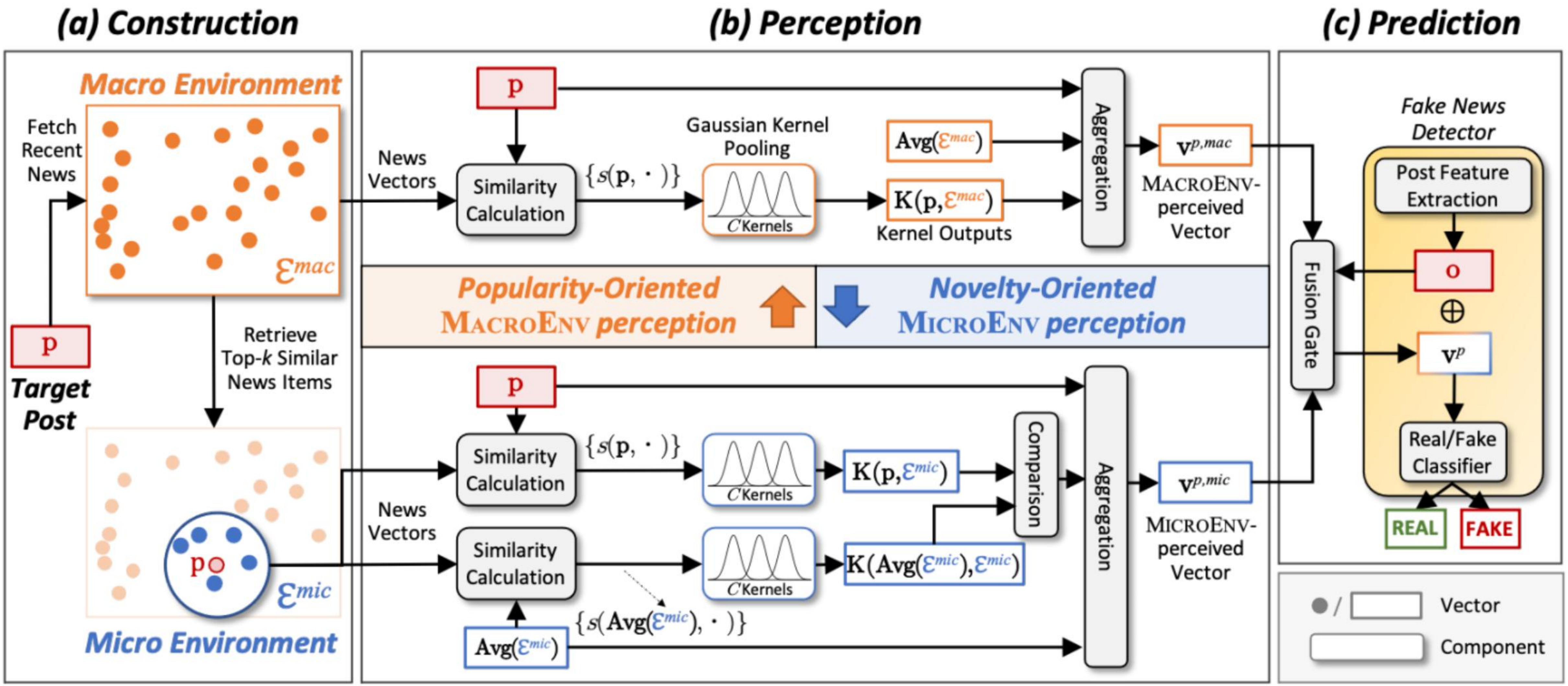


Figure 1: Existing methods for fake news detection rely on (a) the post content itself and (b) related post-level signals like social context and knowledge. Unlike (a) and (b), our method captures (c) signals from *news environments*.

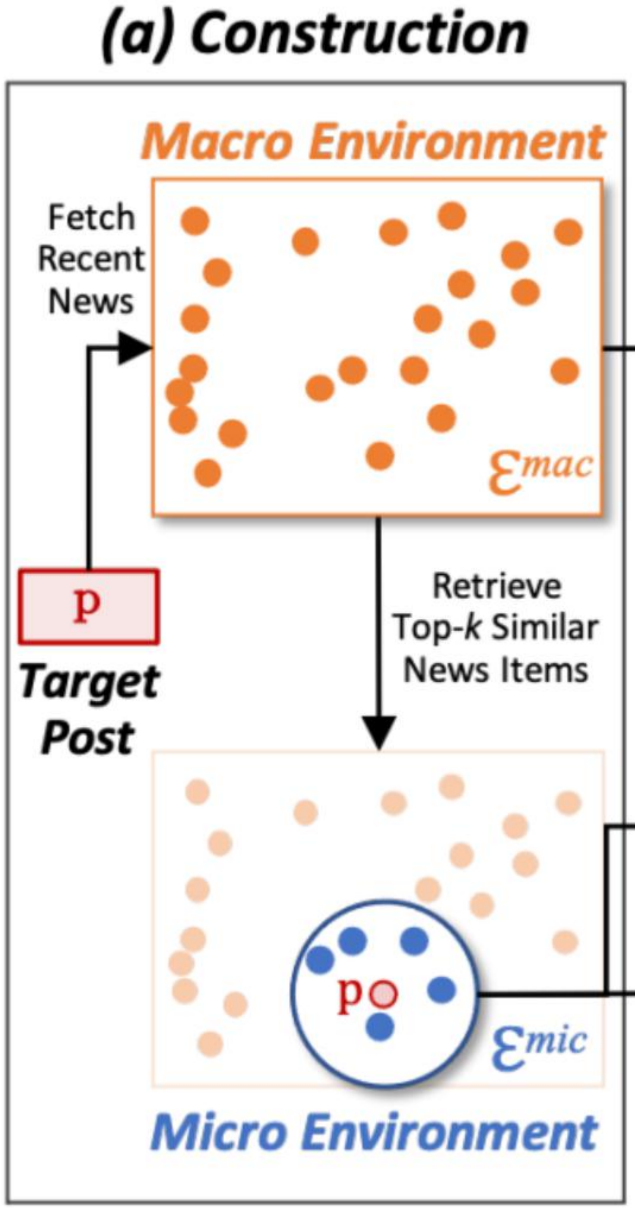
Introduction



Method


(a) Construction
(b) Perception
(c) Prediction
Macro Environment
Fetch Recent News
p
Target Post
Retrieve Top-k Similar News Items
Micro Environment
News Vectors
Similarity Calculation
Gaussian Kernel Pooling
C Kernels
Avg(E^mac)
K(p,E^mac)
Kernel Outputs
Aggregation
v^{p,mac}
MacroEnv-perceived Vector
Popularity-Oriented MacroEnv perception
Novelty-Oriented MicroEnv perception
K(p,E^mic)
K(Avg(E^mic),E^mic)
Avg(E^mic)
Comparison
v^{p,mic}
MicroEnv-perceived Vector
Fake News Detector
Post Feature Extraction
o
v^p
Fusion Gate
Real/Fake Classifier
REAL
FAKE
Vector
Component

Method



$$\mathcal{E}^{mac} = \{e : e \in \mathcal{E}, 0 < t_p - t_e \le T\}, \quad (1)$$

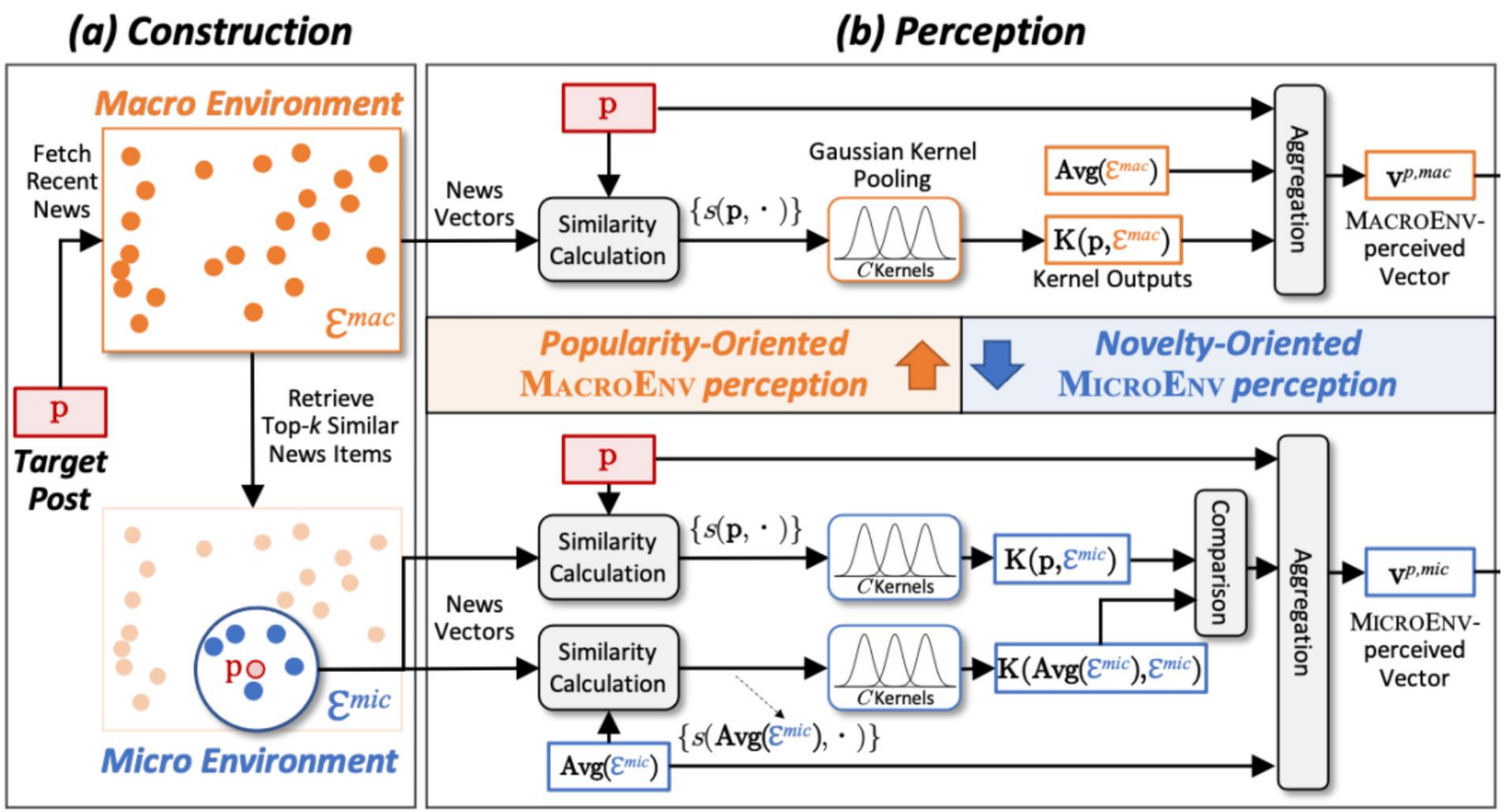
where t_p and t_e respectively denote the publication date of p and the news item e.

$$\mathcal{E}^{mic} = \{e : e \in \text{Topk}(p, \mathcal{E}^{mac})\}, \quad (2)$$

where $k = \lceil r|\mathcal{E}^{mac}| \rceil$ and $r \in (0, 1)$ determines the proportion.

$$\mathbf{p} = \mathcal{M}(p), \ \ \mathbf{e} = \mathcal{M}(e). \quad (3)$$

Method



$$s(\mathbf{p}, \mathbf{e}_i) = \frac{\mathbf{p} \cdot \mathbf{e}_i}{\|\mathbf{p}\| \|\mathbf{e}_i\|}. \quad (4)$$

$$\mathbf{K}_k^i = \exp\left(-\frac{(s(\mathbf{p}, \mathbf{e}_i) - \mu_k)^2}{2\sigma_k^2}\right), \quad (5)$$

$$\mathbf{K}_k(\mathbf{p}, \mathcal{E}^{mac}) = \sum_{i=1}^{|\mathcal{E}^{mac}|} \mathbf{K}_k^i, \quad (6)$$

$$\mathbf{K}(\mathbf{p}, \mathcal{E}^{mac}) = \mathrm{Norm}\left(\bigoplus_{k=1}^{C} \mathbf{K}_k(\mathbf{p}, \mathcal{E}^{mac})\right), \quad (7)$$

$$\mathbf{v}^{p,mac} = \mathrm{MLP}(\mathbf{p} \oplus \mathbf{m}(\mathcal{E}^{mac}) \oplus \mathbf{K}(\mathbf{p}, \mathcal{E}^{mac})). \quad (8)$$

Method

Specifically, we employ a Gaussian Kernel Pooling proposed in (Xiong et al., 2017) across the range of cosine similarity to get soft counting values. Assuming that we use C kernels $\{\mathbf{K}_i\}_{i=1}^{C}$, the output of k-th kernel is:

$$\mathbf{K}_k^i = \exp\left(-\frac{(s(\mathbf{p}, \mathbf{e}_i) - \mu_k)^2}{2\sigma_k^2}\right), \tag{5}$$

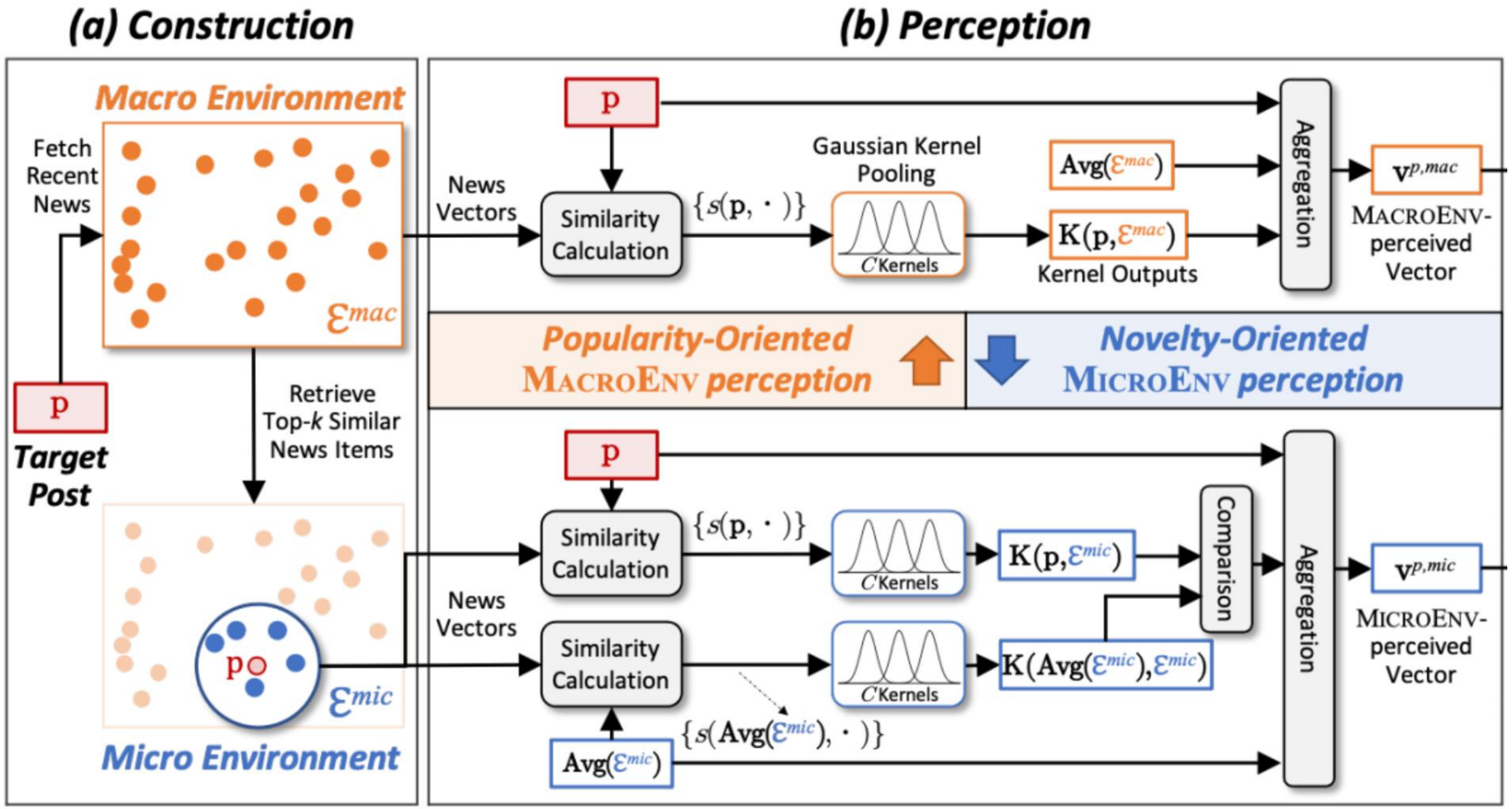
$$\mathbf{K}_k(\mathbf{p}, \mathcal{E}^{mac}) = \sum_{i=1}^{|\mathcal{E}^{mac}|} \mathbf{K}_k^i, \tag{6}$$

where μ_k and σ_k is the mean and width of the k-th kernel. In Eq. (5), if the similarity between $\mathbf{p}$ and $\mathbf{e}$ is close to μ_k, the exponential term will be close to 1; otherwise to 0. We then sum the exponential terms with Eq. (6). This explains why a kernel is like a soft counting bin of similarities. We here scatter the means $\{\mu_k\}_{k=1}^{C}$ of the C kernels in $[-1, 1]$ to completely and evenly cover the range of cosine similarity. The widths are controlled by $\{\sigma_k\}_{k=1}^{C}$. Appendix B.1 provides the details. A

B.1 Kernel Settings

We use $C = 22$ kernels for softly counting the cosine similarities. Following (Xiong et al., 2017), we first determine 21 kernels whose μs scatter in $[-1, 1]$ with an interval of 0.1 and σ^2s are all 0.05. Then we add a kernel with a μ of 0.99 and a σ^2 of 0.01, specially for extremely similar situations. The final kernel list is [(-1.0, 0.1), (-0.9, 0.1),$\cdots$, (1.0,0.1), (0.99, 0.01)]

Method



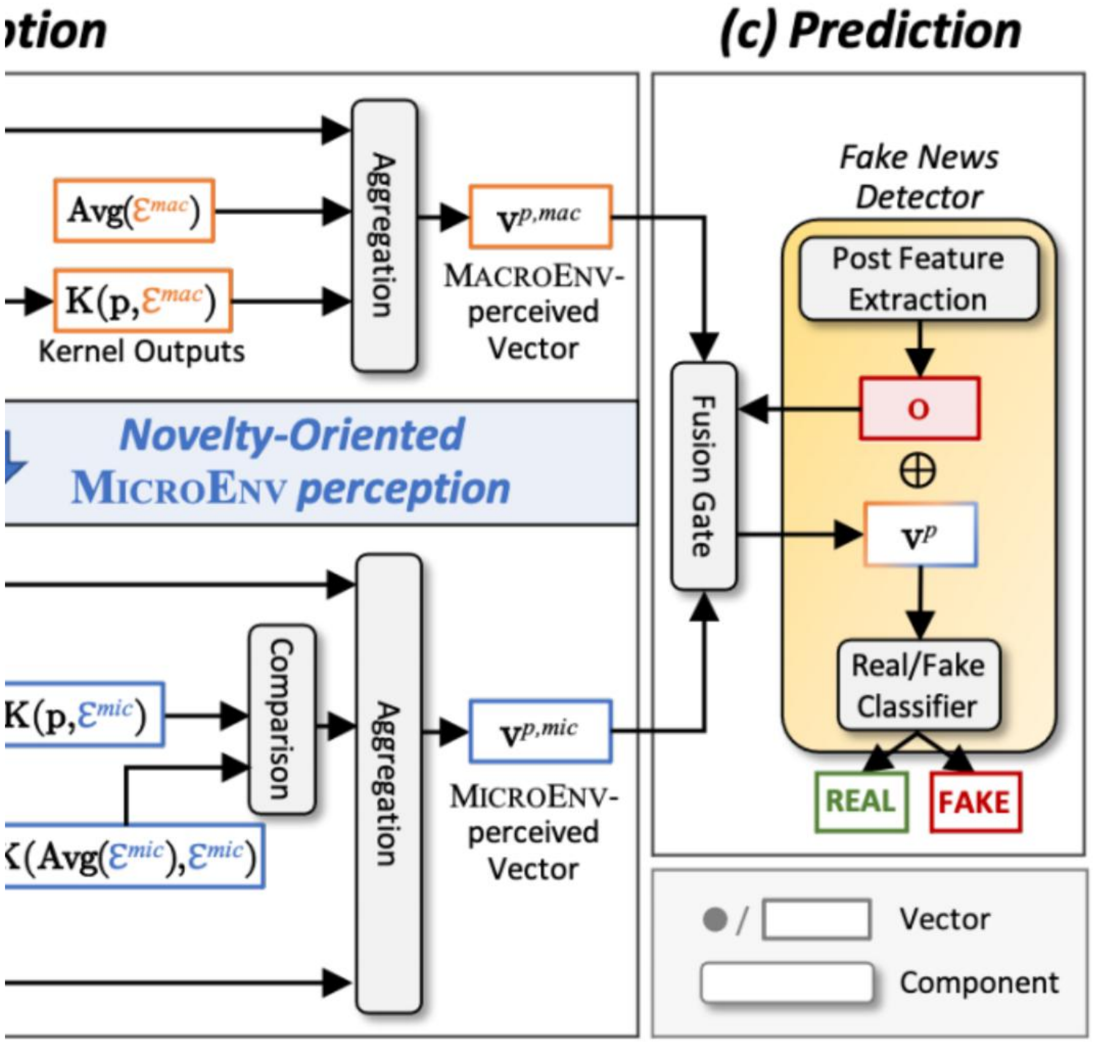
$$\mathbf{u}^{sem} = \mathrm{MLP}(\mathbf{p} \oplus \mathbf{m}(\mathcal{E}^{mic})), \tag{9}$$

$$\mathbf{u}^{sim} = \mathrm{MLP}(\mathrm{g}(\mathbf{K}(\mathbf{p}, \mathcal{E}^{mic}), \mathbf{K}(\mathbf{m}(\mathcal{E}^{mic}), \mathcal{E}^{mic}))) \tag{10}$$

$$\mathbf{v}^{p,mic} = \mathrm{MLP}(\mathbf{u}^{sem} \oplus \mathbf{u}^{sim}), \tag{11}$$

where the comparison function $\mathrm{g}(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \odot \mathbf{y}) \oplus (\mathbf{x} - \mathbf{y})$ and $\odot$ is the Hadamard product operator. $\mathbf{u}^{sem}$ and $\mathbf{u}^{sim}$ respectively aggregate the semantic and similarity information. The MLPs are individually parameterized. We omit their index numbers in the above equations for brevity.

Method



$$\mathbf{v}^p = \mathbf{g} \odot \mathbf{v}^{p,mac} + (\mathbf{1} - \mathbf{g}) \odot \mathbf{v}^{p,mic}, \qquad (12)$$

where the gating vector $\mathbf{g} = \text{sigmoid}(\text{Linear}(\mathbf{o} \oplus \mathbf{v}^{p,mac}))$, sigmoid is to constrain the value of each element in $[0, 1]$, and $\mathbf{o}$ denotes the last-layer feature from a post-only detector.[3]

$$\hat{\mathbf{y}} = \text{softmax}(\text{MLP}(\mathbf{o} \oplus \mathbf{v}^p)). \qquad (13)$$

When working with more complex detectors that rely on other sources besides the post, we can simply concatenate those feature vectors in Eq. (13). For example, we can concatenate $\mathbf{v}^p$ with the post-article joint representation if the fake news detector is knowledge-based. During training, we minimize the cross-entropy loss.

Experiments

Table 1: Statistics of the datasets.

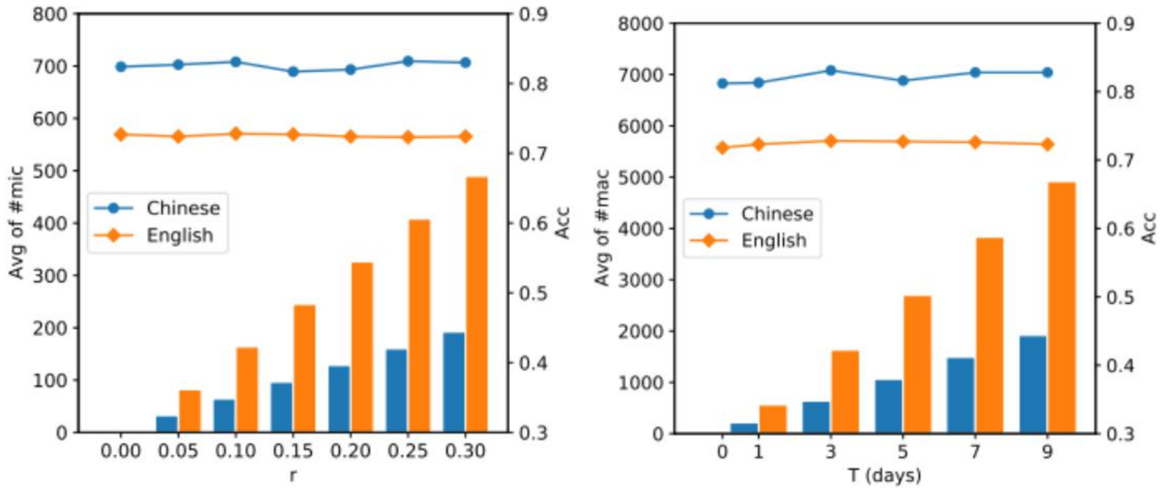
Dataset	**Chinese**			**English**		
	Train	Val	Test	Train	Val	Test
#Real	8,787	5,131	5,625	1,976	656	661
#Fake	8,992	4,923	5,608	1,924	638	628
Total	17,779	10,054	11,233	3,900	1,294	1,289
#News Items	583,208			1,003,646		
Min/Avg/Max of $\lvert\mathcal{E}^{mac}\rvert$ in 3 days	41 / 505 / 1,563			308 / 1,614 / 2,211		

Experiments

Table 2: Performance comparison of base models with and without the NEP. The better result in each group using the same base model are in **boldface**.

Model		Chinese				English			
		Acc.	macF1	$F1_{fake}$	$F1_{real}$	Acc.	macF1	$F1_{fake}$	$F1_{real}$
Post-Only	Bi-LSTM	0.727	0.713	0.652	0.775	0.705	0.704	0.689	**0.719**
	+*NEP*	**0.776**	**0.771**	**0.739**	**0.803**	**0.718**	**0.718**	**0.720**	0.716
	$EANN_T$	0.732	0.718	0.657	0.780	0.700	0.699	0.683	0.714
	+*NEP*	**0.776**	**0.770**	**0.733**	**0.807**	**0.722**	**0.722**	**0.722**	**0.722**
	BERT	0.792	0.785	0.744	0.825	0.709	0.709	0.701	**0.716**
	+*NEP*	**0.810**	**0.805**	**0.772**	**0.837**	**0.718**	**0.718**	**0.720**	0.715
	BERT-Emo	0.812	0.807	0.776	0.838	0.718	0.718	0.719	0.718
	+*NEP*	**0.831**	**0.829**	**0.808**	**0.850**	**0.728**	**0.728**	**0.728**	**0.728**
"Zoom-In"	DeClarE	0.764	0.758	0.720	0.795	0.714	0.714	0.709	**0.718**
	+*NEP*	**0.800**	**0.797**	**0.773**	**0.822**	**0.717**	**0.716**	**0.718**	0.714
	MAC	0.755	0.751	0.717	0.784	0.706	0.705	0.708	0.701
	+*NEP*	**0.764**	**0.760**	**0.732**	**0.789**	**0.716**	**0.716**	**0.716**	**0.716**

Experiments



(a) Proportion Factor r (b) Day Difference T

Figure 4: Effects of (a) the proportion factor r and (b) the day difference T. Lines show the accuracies and bars show the average numbers of news items in the micro/macro environments.

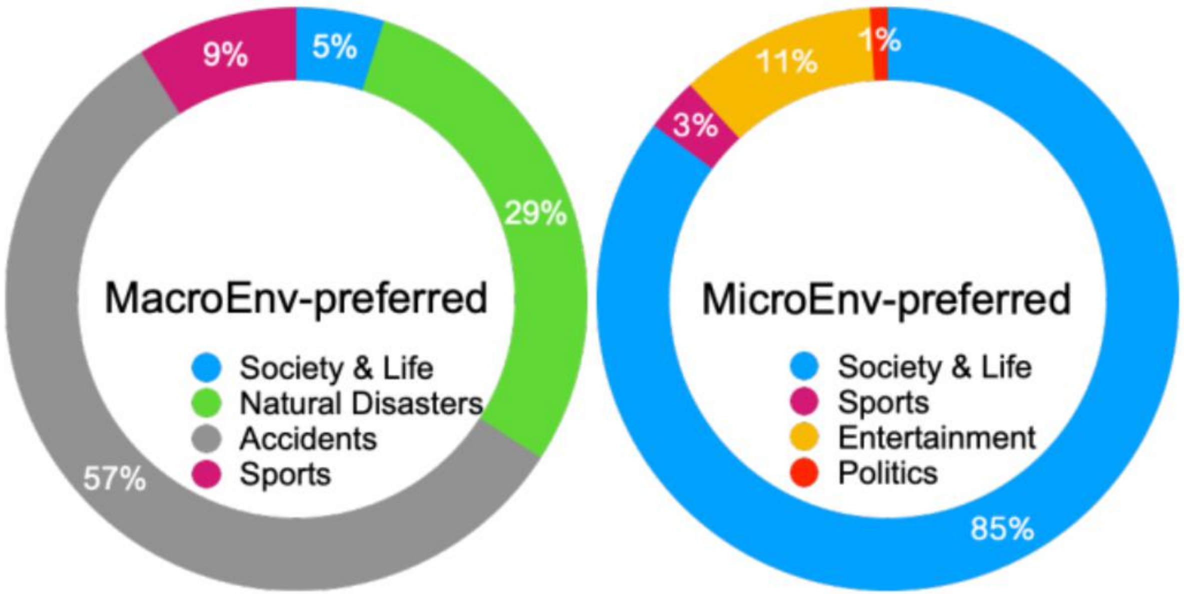


Figure 5: Categories of MACROENV- and MICROENV-preferred samples.

Experiments

Table 3: Performance comparison of the NEP and its variants without the fake news detector or without the environment perception module. The best result in each group is in **boldface**.

Model	Chinese				English			
	Acc.	macF1	$F1_{fake}$	$F1_{real}$	Acc.	macF1	$F1_{fake}$	$F1_{real}$
MacroEnv	0.689	0.659	0.557	0.761	0.693	0.693	**0.696**	0.689
MicroEnv	0.666	0.626	0.503	0.748	0.695	0.695	0.694	0.696
MacroEnv+MicroEnv	**0.694**	**0.666**	**0.569**	**0.763**	**0.696**	**0.696**	0.694	**0.697**
BERT-Emo + *NEP*	**0.831**	**0.829**	**0.808**	**0.850**	**0.728**	**0.728**	**0.728**	0.728
w/o MacroEnv	0.822	0.819	0.794	0.843	0.726	0.726	0.726	0.725
w/o MicroEnv	0.824	0.820	0.795	0.845	0.723	0.723	0.715	**0.731**
DeClarE + *NEP*	**0.797**	**0.800**	**0.773**	**0.822**	**0.717**	**0.716**	0.718	**0.714**
w/o MacroEnv	0.776	0.771	0.735	0.806	0.712	0.711	0.709	0.713
w/o MicroEnv	0.778	0.773	0.736	0.809	0.709	0.709	**0.719**	0.698

Experiments

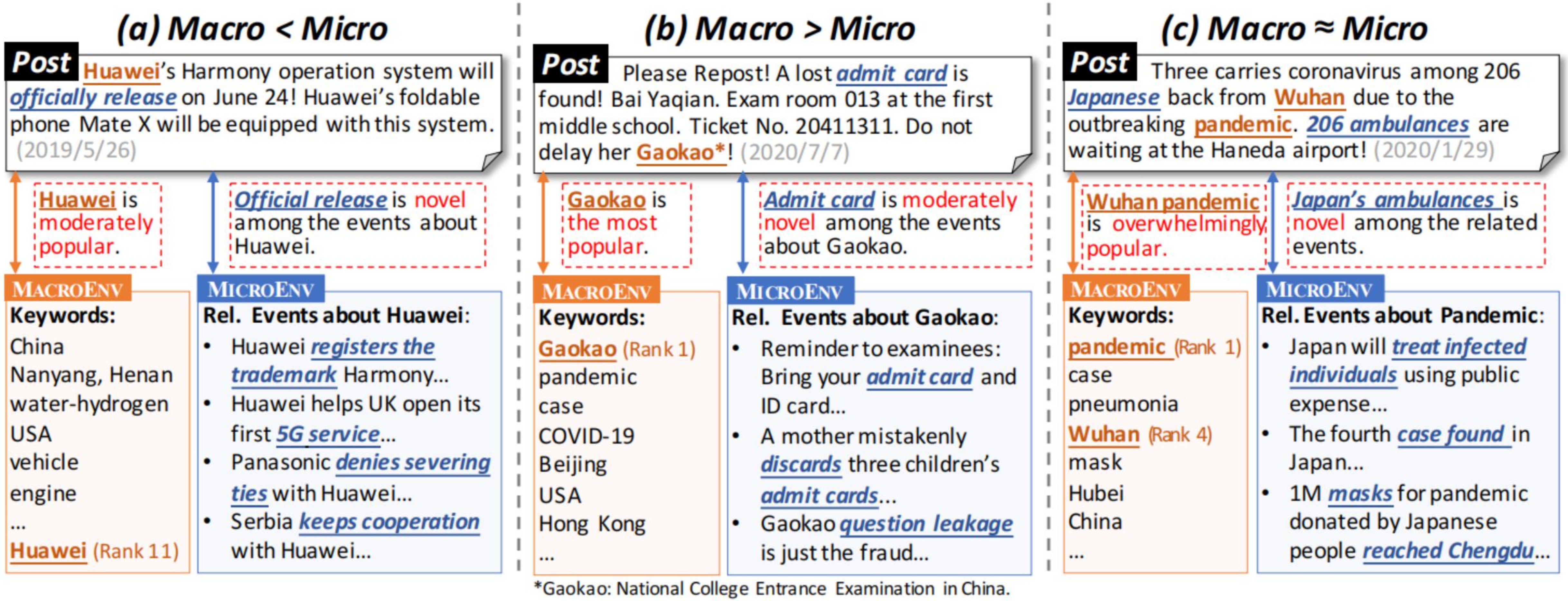


Figure 6: Three fake news cases with different preferences on environmental information. Underlined regular words hit the keywords in the MacroEnv and *underlined italic* words are related to the MicroEnv. Keywords are extracted using TextRank (Mihalcea and Tarau, 2004).

Experiments

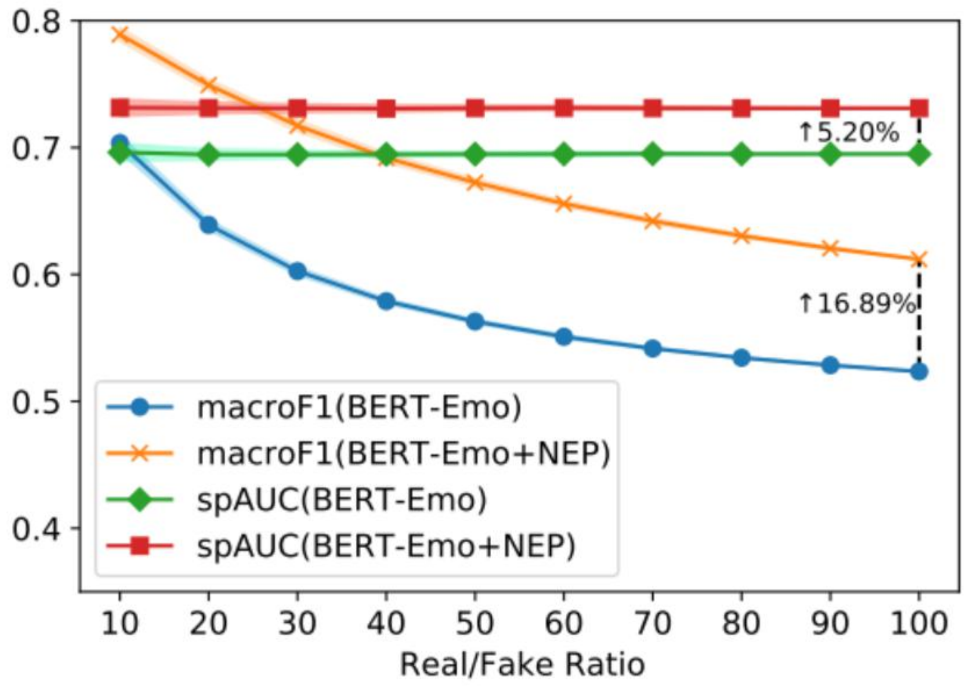


Figure 7: Macro F1s and spAUCs on the online data in different real/fake ratios. We sampled 100 times from the 100:1 set for each fo the first nine ratios. Shadows show the standard deviations. The percentages denote relative improvements using the NEP.

Thanks