Context-aware Entity Morph Decoding

Boliang Zhang[1], Hongzhao Huang[1], Xiaoman Pan[1], Sujian Li[2], Chin-Yew Lin[3]
Heng Ji[1], Kevin Knight[4], Zhen Wen[5], Yizhou Sun[6], Jiawei Han[7], Bulent Yener[1]
[1]Rensselaer Polytechnic Institute, [2]Peking University, [3]Microsoft Research Asia, [4]University of Southern California
[5]IBM T. J. Watson Research Center, [6]Northeastern University, [7]Univerisity of Illinois at Urbana-Champaign
[1]{zhangb8,huangh9,panx2,jih,yener}@rpi.edu, [2]lisujian@pku.edu.cn, [3]cyl@microsoft.com
[4]hanj@illinois.edu, [5]zhenwen@us.ibm.com, [6]yzsun@ccs.neu.edu, [7]hanj@illinois.edu

Abstract

People create morphs, a special type of fake alternative names, to achieve certain communication goals such as expressing strong sentiment or evading censors. For example, "*Black Mamba*", the name for a highly venomous snake, is a morph that *Kobe Bryant* created for himself due to his agility and aggressiveness in playing basketball games. This paper presents the first end-to-end context-aware entity morph decoding system that can automatically identify, disambiguate, verify morph mentions based on specific contexts, and resolve them to target entities. Our approach is based on an absolute "*cold-start*" - it does not require any candidate morph or target entity lists as input, nor any manually constructed morph-target pairs for training. We design a semi-supervised collective inference framework for morph mention extraction, and compare various deep learning based approaches for morph resolution. Our approach achieved significant improvement over the state-of-the-art method (Huang et al., 2013), which used a large amount of training data. [1]

1 Introduction

Morphs (Huang et al., 2013; Zhang et al., 2014) refer to the fake alternative names created by social media users to entertain readers or evade censors. For example, during the World Cup in 2014, a morph "*Su-tooth*" was created to refer to the Uruguay striker "*Luis Suarez*" for his habit of biting other players. Automatically decoding human-generated morphs in text is critical for downstream deep language understanding tasks such as entity linking and event argument extraction.

However, even for human, it is difficult to decode many morphs without certain historical, cultural, or political background knowledge (Zhang et al., 2014). For example, "*The Hutt*" can be used to refer to a fictional alien entity in the *Star Wars* universe ("*The Hutt stayed and established himself as ruler of Nam Chorios*"), or the governor of New Jersey, *Chris Christie* ("*The Hutt announced a bid for a seat in the New Jersey General Assembly*"). Huang et al. (2013) did a pioneering pilot study on morph resolution, but their approach assumed the entity morphs were already extracted and used a large amount of labeled data. In fact, they resolved morphs on corpus-level instead of mention-level and thus their approach was context-independent. A practical morph decoder, as depicted in Figure 1, consists of two problems: (1) Morph Extraction: given a corpus, extract morph mentions; and (2). Morph Resolution: For each morph mention, figure out the entity that it refers to.

In this paper, we aim to solve the fundamental research problem of end-to-end morph decoding and propose a series of novel solutions to tackle the following challenges.

Challenge 1: Large-scope candidates

Only a very small percentage of terms can be used as morphs, which should be interesting and fun. As we annotate a sample of $4,668$ Chinese weibo tweets, only 450 out of $19,704$ unique terms are morphs. To extract morph mentions, we propose a

[1]The data set and programs are publicly available at: http://nlp.cs.rpi.edu/data/morphdecoding.zip and http://nlp.cs.rpi.edu/software/morphdecoding.tar.gz

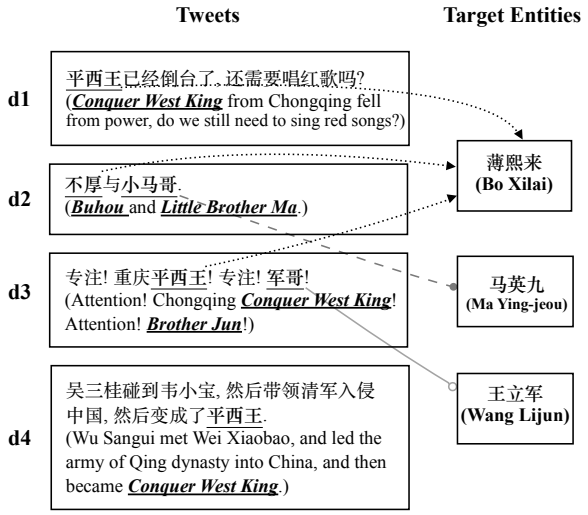

Figure 1: An Illustration of Morph Decoding Task.

two-step approach to first identify individual mention candidates to narrow down the search scope, and then verify whether they refer to morphed entities instead of their original meanings.

Challenge 2: Ambiguity, Implicitness, Informality

Compared to regular entities, many morphs contain informal terms with hidden information. For example, "不厚 *(not thick)*" is used to refer to "薄熙来 *(Bo Xilai)*" whose last name "薄 *(Bo)*" means "thin". Therefore we attempt to model the rich contexts with careful considerations for morph characteristics both globally (e.g., language models learned from a large amount of data) and locally (e.g. phonetic anomaly analysis) to extract morph mentions.

For morph resolution, the main challenge lies in that the surface forms of morphs usually appear quite different from their target entity names. Based on the distributional hypothesis (Harris, 1954) which states that words that often occur in similar contexts tend to have similar meanings, we propose to use deep learning techniques to capture and compare the deep semantic representations of a morph and its candidate target entities based on their contextual clues. For example, the morph "平西王*(Conquer West King)*" and its target entity "薄熙来 *(Bo Xilai)*" share similar implicit contextual representations such as "重庆*(Chongqing)*" (Bo was the governor of Chongqing) and "倒台 *(fall from power)*".

Challenge 3: Lack of labeled data

To the best of our knowledge, no sufficient mention-level morph annotations exist for training an end-to-end decoder. Manual morph annotations require native speakers who have certain cultural background (Zhang et al., 2014). In this paper we focus on exploring novel approaches to save annotation cost in each step. For morph extraction, based on the observation that morphs tend to share similar characteristics and appear together, we propose a semi-supervised collective inference approach to extract morph mentions from multiple tweets simultaneously. Deep learning techniques have been successfully used to model word representation in an unsupervised fashion. For morph resolution, we make use of a large amount of unlabeled data to learn the semantic representations of morphs and target entities based on the unsupervised continuous bag-of-words method (Mikolov et al., 2013b).

2 Problem Formulation

Following the recent work on morphs (Huang et al., 2013; Zhang et al., 2014), we use Chinese Weibo tweets for experiments. Our goal is to develop an end-to-end system that automatically extract morph mentions and resolve them to their target entities. Given a corpus of tweets $D = \{d_1, d_2, ..., d_{|D|}\}$, we define a candidate *morph* m_i as a unique term t_j in T, where $T = \{t_1, t_2, ..., t_{|T|}\}$ is the set of unique terms in D. To extract T, we first apply several well-developed Natural Language Processing tools, including Stanford Chinese word segmenter (Chang et al., 2008), Stanford part-of-speech tagger (Toutanova et al., 2003) and Chinese lexical analyzer ICTCLAS (Zhang et al., 2003), to process the tweets and identify noun phrases. Then we define a *morph mention* m_i^p of m_i as the p-th occurrence of m_i in a specific document d_j. Note that a mention with the same surface form as m_i but referring to its original entity is not considered as a morph mention. For instance, the "平西王 *(Conquer West King)*" in d_1 and d_3 in Figure 1 are morph mentions since they refer to the modern politician "薄熙来 *(Bo Xilai)*", while the one in d_4 is not a morph mention since it refers to the original entity, who was king "吴三桂 *(Wu Sangui)*".

For each morph mention, we discover a list of target candidates $E = \{e_1, e_2, ..., e_{|E|}\}$ from Chinese web data for morph mention resolution. We

design an end-to-end morph decoder which consists of the following procedure:

- ***Morph Mention Extraction***
  - ***Potential Morph Discovery***: This first step aims to obtain a set of potential entity-level morphs $M = \{m_1, m_2, ...\}(M \subseteq T)$. Then, we only verify and resolve the mentions of these potential morphs, instead of all the terms in T in a large corpus.
  - ***Morph Mention Verification***: In this step, we aim to verify whether each mention m_i^p of the potential morph $m_i(m_i \in M)$ from a specific context d_j is a morph mention or not.
- ***Morph Mention Resolution***: The final step is to resolve each morph mention m_i^p to its target entity (e.g., "薄熙来 *(Bo Xilai)*" for the morph mention "平西王 *(Conquer West King)*" in d_1 in Figure 1).

3 Morph Mention Extraction

3.1 Why Traditional Entity Mention Extraction doesn't Work

In order to automatically extract morph mentions from any given documents, our first reflection is to formulate the task as a sequence labeling problem, just like labeling regular entity mentions. We adopted the commonly used conditional random fields (CRFs) (Lafferty et al., 2001) and got only 6% F-score. Many morphs are not presented as regular entity mentions. For example, the morph "天线 *(Antenna)*" refers to "温家宝 *(Wen Jiabao)*" because it shares one character "宝 *(baby)*" with the famous children's television series "天线宝宝 *(Teletubbies)*". Even when they are presented as regular entity mentions, they must refer to new target entities which are different from the regular ones. So we propose the following novel two-step solution.

3.2 Potential Morph Discovery

We first introduce the first step of our approach – potential morph discovery, which aims to narrow down the scope of morph candidates without losing recall. This step takes advantage of the common characteristics shared among morphs and identifies the potential morphs using a supervised method, since it is relatively easy to collect a certain number of corpus-level morphs as training data compared to labeling morph mentions. Through formulating this task as a binary classification problem, we adopt the Support Vector Machines (SVMs) (Cortes and Vapnik, 1995) as the learning model. We propose the following four categories of features.

Basic: (i) character unigram, bigram, trigram, and surface form; (ii) part-of-speech tags; (iii) the number of characters; (iv) whether some characters are identical. These basic features will help identify several common characteristics of morph candidates (e.g., they are very likely to be nouns, and very unlikely to contain single characters).

Dictionary: Many morphs are non-regular names derived from proper names while retaining some characteristics. For example, the morphs "薄督 *(Governor Bo)*" and "吃省 *(Gourmand Province)*" are derived from their target entity names "薄熙来 *(Bo Xilai)*" and "广东省 *(Guandong Province)*", respectively. Therefore, we adopt a dictionary of proper names (Li et al., 2012) and propose the following features: (i) Whether a term occurs in the dictionary. (ii) Whether a term starts with a commonly used last name, and includes uncommonly used characters as its first name. (iii) Whether a term ends with a geo-political entity or organization suffix word, but it's not in the dictionary.

Phonetic: Many morphs are created based on phonetic (Chinese pinyin in our case) modifications. For instance, the morph "饭饼饼 *(Rice Cake)*" has the same phonetic transcription as its target entity name "范冰冰 *(Fan Bingbing)*". To extract phonetic-based features, we compile a dictionary composed of ⟨phonetic transcription, term⟩ pairs from the Chinese Gigaword corpus [2]. Then for each term, we check whether it has the same phonetic transcription as any entry in the dictionary but they include different characters.

Language Modeling: Many morphs rarely appear in a general news corpus (e.g., "六步郎 *(Six Step Man)*" refers to the NBA baseketball player "勒布朗·詹姆斯 *(Lebron James)*".). Therefore, we use the character-based language models trained from Gigaword to calculate the occurrence probabilities of each term, and use n-gram probabilities ($n \in [1:5]$) as features.

3.3 Morph Mention Verification

The second step is to verify whether a mention of the discovered potential morphs is indeed used as a morph in a specific context. Based on the ob-

[2] https://catalog.ldc.upenn.edu/LDC2011T07

servation that closely related morph mentions often occur together, we propose a semi-supervised graph-based method to leverage a small set of labeled seeds, coreference and correlation relations, and a large amount of unlabeled data to perform collective inference and thus save annotation cost. According to our observation of morph mentions, we propose the following two hypotheses:

Hypothesis 1: *If two mentions are coreferential, then they both should either be morph mentions or non-morph mentions.* For instance, the morph mentions "平西王 *(Conquer West King)*" in d_1 and d_3 in Figure 1 are coreferential, they both refer to the modern politician "薄熙来 *(Bo Xilai)*".

Hypothesis 2: *Those highly correlated mentions tend to either be morph mentions or non-morph mentions.* From our annotated dataset, 49% morph mentions co-occur on tweet level. For example, "平西王*(Conquer West King)*" and "军哥*(Brother Jun)*" are used together in d_3 in Figure 1.

Based on these hypotheses, we aim to design an effective approach to compensate for the limited annotated data. Graph-based semi-supervised learning approaches (Zhu et al., 2003; Smola and Kondor, 2003; Zhou et al., 2004) have been successfully applied many NLP tasks (Niu et al., 2005; Chen et al., 2006; Huang et al., 2014). Therefore we build a mention graph to capture the semantic relatedness (weighted arcs) between potential morph mentions (nodes) and propose a semi-supervised graph-based algorithm to collectively verify a set of relevant mentions using a small amount of labeled data. We now describe the detailed algorithm as follows.

Mention Graph Construction

First, we construct a mention graph that can reflect the association between all the mentions of potential morphs. According to the above two hypotheses, *mention coreference* and *correlation* relations are the basis to build our mention graph, which is represented by a matrix.

In Chinese Weibo, their exist rich and clean social relations including *authorship*, *replying*, *retweeting*, or *user mentioning* relations. We make use of these social relations to judge the possibility of two mentions of the same potential morph being coreferential. If there exists one social relation between two mentions m_i^p and m_i^q of the morph m_i, they are usually coreferential and assigned an association score 1. We also detect coreferential relations by performing content similarity analysis. The cosine similarity is adopted with the tf-idf representation for the contexts of two mentions. Then we get a coreference matrix W^1:

$$W^1_{m_i^p,m_i^q} = \begin{cases} 1.0 & \text{if } m_i^p \text{ and } m_i^q \text{ are linked with certain social relation} \\ cos(m_i^p, m_i^q) & \text{else if } q \in kNN(p) \\ 0 & \text{Otherwise} \end{cases}$$

where m_i^p and m_i^q are two mentions from the same potential morph m_i, and kNN means that each mention is connected to its k nearest neighboring mentions.

Users tend to use morph mentions together to achieve their communication goals. To incorporate such evidence, we measure the correlation between two mentions m_i^p and m_j^q of two different potential morphs m_i and m_j as $corr(m_i^p, m_j^q) = 1.0$ if there exists a certain social relation between them. Otherwise, $corr(m_i^p, m_j^q) = 0$. Then we can obtain the correlation matrix: $W^2_{m_i^p,m_j^q} = corr(m_i^p, m_j^q)$.

To tune the balance of coreferential relation and correlation relation during learning, we first get two matrices $\hat{W}^1$ and $\hat{W}^2$ by row-normalizing W^1 and W_2, respectively. Then we obtain the final mention matrix W with a linear combination of $\hat{W}^1$ and $\hat{W}^2$: $W = \alpha\hat{W}^1 + (1-\alpha)\hat{W}^2$, where α is the coefficient between 0 and 1 [3].

Graph-based Semi-supervised Learning

Intuitively, if two mentions are strongly connected, they tend to hold the same label. The label of 1 indicates a mention is a morph mention, and 0 means a non-morph mention. We use $Y = \begin{bmatrix} Y_l & Y_u \end{bmatrix}^T$ to denote the label vector of all mentions, where the first l nodes are verified mentions labeled as 1 or 0, and the remaining u nodes need to be verified and initialized with the label 0.5. Our final goal is to obtain the final label vector Y_u by incorporating evidence from initial labels and the mention graph.

Following the graph-based semi-supervised learning algorithm (Zhu et al., 2003), the mention verification problem is formulated to optimize the objective function $\mathcal{Q}(\mathcal{Y}) = \mu \sum_{i=1}^{l}(y_i - y_i^0)^2 + \frac{1}{2}\sum_{i,j} W_{ij}(y_i - y_j)^2$ where y_i^0 denotes the initial

[3] α is set to 0.8 in this paper, optimized from the development set.

label, and μ is a regularization parameter that controls the trade-off between initial labels and the consistency of labels on the mention graph. Zhu et al. (2003) has proven that this formula has both closed-form and iterative solutions.

4 Morph Mention Resolution

The final step is to resolve the extracted morph mentions to their target entities.

4.1 Candidate Target Identification

We start from identifying a list of target candidates for each morph mention from the comparable corpora including Sina Weibo, Chinese News and English Twitter. After preprocessing the corpora using word segmentation, noun phrase chunking and name tagging, the name entity list is still too large and too noisy for candidate ranking. To clean the name entity list, we adopt the temporal Distribution Assumption proposed in our recent work (Huang et al., 2013). It assumes that a morph m and its real target e should have similar temporal distributions in terms of their occurrences. Following the same heuristic we assume that an entity is a valid candidate for a morph if and only if the candidate appears fewer than seven days after the morph's appearance.

4.2 Candidate Target Ranking

Motivations of Using Deep Learning

Compared to regular entity linking tasks (Ji et al., 2010; Ji et al., 2011; Ji et al., 2014), the major challenge of ranking a morph's candidate target entities lies in that the surface features such as the orthographic similarity between morph and target candidates have been proven inadequate (Huang et al., 2013). Therefore, it is crucial to capture the semantics of both mentions and target candidates. For instance, in order to correctly resolve "平西王 *(Conquer West King)*" from d_1 and d_3 in Figure 1 to the modern politician "薄熙来*(Bo Xilai)*" instead of the ancient king "吴三桂 *(Wu Sangui)*", it is important to model the surrounding contextual information effectively to capture important information (e.g., "重庆 *(Chongqing)*", "倒台 *(fall from power)*", and "唱红歌 *(sing red songs)*") to represent the mentions and target entity candidates. Inspired by the recent success achieved by deep learning based techniques on learning semantic representations for various NLP tasks (e.g., (Bengio et al., 2003; Collobert et al., 2011; Mikolov et al., 2013b; He et al., 2013)), we design and compare the following two approaches to employ hierarchical architectures with multiple hidden layers to extract useful features and map morphs and target entities into a latent semantic space.

Pairwise Cross-genre Supervised Learning

Ideally, we hope to obtain a large amount of coreferential entity mention pairs for training. A natural knowledge resource is Wikipedia which includes anchor links. We compose an anchor's surface string and the title of the entity it's linked to as a positive training pair. Then we randomly sample negative training instances from those pairs that don't share any links.

Our approach consists of the following steps: (1) generating high quality embedding for each training instance; (2) pre-training with the stacked denoising auto-encoder (Bengio et al., 2003) for feature dimension reduction; and (3) supervised fine-tuning to optimize the neural networks towards a similarity measure (e.g., dot product). Figure 2 depicts the overall architecture of this approach.

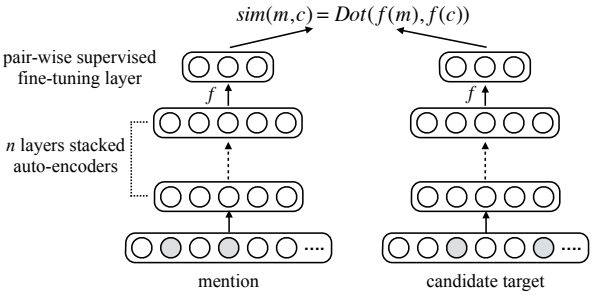

Figure 2: Overall Architecture of Pairwise Cross-genre Supervised Learning

However, morph resolution is significantly different from the traditional entity linking task since the latter mainly focuses on formal and explicit entities (e.g., "薄熙来 *(Bo Xilai)*") which tend to have stable referents in Wikipedia. In contrast, morphs tend to be informal, implicit and have newly emergent meanings which evolve over time. In fact, these morph mentions rarely appear in Wikipedia. For example, almost all "平西王 *(Conquer West King)*" mentions in Wikipedia refer to the ancient king instead of the modern politician "薄熙来 *(Bo Xilai)*". In addition, the contextual words in Wikipedia used to describe entities are quite different from those in social media. For example, to describe a death event, Wikipedia usu-

ally uses a formal expression "去世 (pass away)" while an informal expression "挂了 (hang up)" is used more often in tweets. Therefore this approach suffers from the knowledge discrepancy between these two genres.

Within-genre Unsupervised Learning

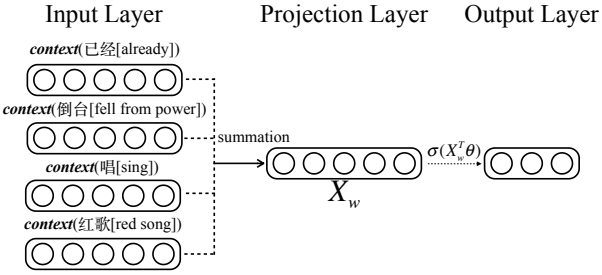

Figure 3: Continuous Bag-of-Words Architecture

To address the above challenge, we propose the second approach to learn semantic embeddings of both morph mentions and entities directly from tweets. Also we prefer unsupervised learning methods due to the lack of training data. Following (Mikolov et al., 2013a), we develop a continuous bag-of-words (CBOW) model that can effectively model the surrounding contextual information. CBOW is discriminatively trained by maximizing the conditional probability of a term w_i given its contexts $c(w_i) = \{w_{i-n}, ..., w_{i-1}, w_{i+1}, ..., w_{i+n}\}$, where n is the contextual window size, and w_i is a term obtained using the preprocessing step introduced in Section 2 [4]. The architecture of CBOW is depicted in Figure 3. We obtain a vector X_{w_i} through the projection layer by summing up the embedding vectors of all terms in $c(w_i)$, and then use the sigmoid activation function to obtain the final embedding of w_i in $c(w_i)$ in the output layer.

Formally, the objective function of CBOW can be formulated as $\mathcal{L}(\theta) = \sum_{w_i \in W} \sum_{w_j \in W} \log p(w_j|c(w_i))$, where W is the set of unique terms obtained from the whole training corpus. $p(w_j|c(w_i))$ is the conditional likelihood of w_j given the context $c(w_i)$ and it is formulated as follows:

$$p(w_j|c(w_i)) = [\sigma(X_{w_i}^T \theta^{w_j})]^{L^{w_i}(w_j)} \times [1 - \sigma(X_{w_i}^T \theta^{w_j})]^{1-L^{w_i}(w_j)},$$

[4]Each w_i is not limited to noun phrases we consider as candidate morphs.

Data	Training	Development	Testing
# Tweets	1,500	500	2,688
# Unique Terms	10,098	4, 848	15,108
# Morphs	250	110	341
# Morph Mentions	1,342	487	2,469

Table 1: Data Statistics

where $L^{w_i}(w_j) = \begin{cases} 1, & w_i = w_j \\ 0, & Otherwise \end{cases}$, σ is the sigmoid activation function, and θ^{w_i} is the embeddings of w_i to be learned with back-propagation during training.

5 Experiments

5.1 Data

We retrieved 1,553,347 tweets from Chinese Sina Weibo from May 1 to June 30, 2013 and 66, 559 web documents from the embedded URLs in tweets for experiments. We then randomly sampled $4,688$ non-redundant tweets and asked two Chinese native speakers to manually annotate morph mentions in these tweets. The annotated dataset is randomly split into training, development, and testing sets, with detailed statistics shown in Table 1 [5]. We used 225 positive instances and 225 negative instances to train the model in the first step of potential morph discovery.

We collected a Chinese Wikipedia dump of October 9th, 2014, which contains 2,539,355 pages. We pulled out person, organization and geo-political pages based on entity type matching with DBpedia [6]. We also filter out the pages with fewer than 300 words. For training the model, we use 60,000 mention-target pairs along with one negative sample randomly generated for each pair, among which, 20% pairs are reserved for parameter tuning.

5.2 Overall: End-to-End Decoding

In this subsection, we first study the end-to-end decoding performance of our best system, and compare it with the state-of-the-art supervised learning-to-rank approach proposed by (Huang et al., 2013) based on information networks construction and traverse with meta-paths. We use the 225 extracted morphs as input to feed (Huang et al., 2013) system. The experiment setting, implementation and evaluation process are similar to (Huang et al., 2013).

[5]We will make all of these annotations and other resources available for research purposes if this paper gets accepted.

[6]http://dbpedia.org

The overall performance of our approach using within-genre learning for resolution is shown in Table 2. We can see that our system achieves significantly better performance (95.0% confidence level by the Wilcoxon Matched-Pairs Signed-Ranks Test) than the approach proposed by (Huang et al., 2013). We found that (Huang et al., 2013) failed to resolve many unpopular morphs (e.g., "小马 *(Little Ma)*" is a morph referring to Ma Yingjiu, and it only appeared once in the data), because it heavily relies on aggregating contextual and temporal information from multiple instances of each morph. In contrast, our unsupervised resolution approach only leverages the pre-trained word embeddings to capture the semantics of morph mentions and entities.

Model	Precision	Recall	F_1
Huang et al., 2013	40.2	33.3	36.4
Our Approach	**41.1**	**35.9**	**38.3**

Table 2: End-to-End Morph Decoding (%)

5.3 Diagnosis: Morph Mention Extraction

The first step discovered 888 potential morphs (80.1% of all morphs, 5.9% of all terms), which indicates that this step successfully narrowed down the scope of candidate morphs.

Method	Precision	Recall	F_1
Naive	58.0	83.1	68.3
SVMs	61.3	80.7	69.7
Our Approach	88.2	77.2	**82.3**

Table 3: Morph Mention Verification (%)

Now we evaluate the performance of morph mention verification. We compare our approach with two baseline methods: (i) *Naive*, which considers all mentions as morph mentions; (ii) *SVMs*, a fully supervised model using Support Vector Machines (Cortes and Vapnik, 1995) based on unigrams and bigrams features. Table 3 shows the results. We can see that our approach achieves significantly better performance than the baseline approaches. In particular it can verify the mentions of newly emergent morphs. For instance, "棒棒棒 *(Good Good Good)*" is mistakenly identified by the first step as a potential morph, but the second step correctly filters it out.

5.4 Diagnosis: Morph Mention Resolution

The target candidate identification step successfully filters 86% irrelevant entities with high precision (98.5% of morphs retain their target entitis). For candidate ranking, we compare with several baseline approaches as follows:

- ***BOW***: We compute cosine similarity over bag-of-words vectors with tf-idf values to measure the context similarity between a mention and its candidates.
- ***Pair-wise Cross-genre Supervised Learning***: We first construct a vocabulary by choosing the top 100,000 frequent terms. Then we randomly sample 48,000 instances for training and 12,000 instances for development. At the pre-training step, we set the number of hidden layers as 3, the size of each hidden layer as 1000, the masking noise probability for the first layer as 0.7, and a Gaussian noise with standard deviation of 0.1 for higher layers. The learning rate is set to be 0.01. At the fine-tuning stage, we add a 200 units layer on top of auto-encoders and optimize the neural network models based on the training data.
- ***Within-genre Unsupervised Learning***: We directly train morph mention and entity embeddings from the large-scale tweets and web documents that we collect. We set the window size as 10 and the vector dimension as 800 based on the development set.

The overall performance of various resolution approaches using perfect morph mentions is shown in Figure 4. We can clearly see that our second within-genre learning approach achieves the best performance. Figure 5 demonstrates the differences between our two deep learning based methods. When learning semantic embeddings directly from Wikipedia, we can see that the top 10 closest entities of the mention "平西王*(Conquer West King)*" are all related to the ancient king "吴三桂*(Wu Sangui)*". Therefore this method is only able to capture the original meanings of morphs. In contrast, when we learn embeddings directly from tweets, most of the closest entities are relevant to its target entity "薄熙来 *(Bo Xilai)*".

6 Related Work

The first morph decoding work (Huang et al., 2013) assumed morph mentions are already discovered and didn't take contexts into account. To the best of our knowledge, this is the first work on context-aware end-to-end morph decoding.

Morph decoding is related to several traditional

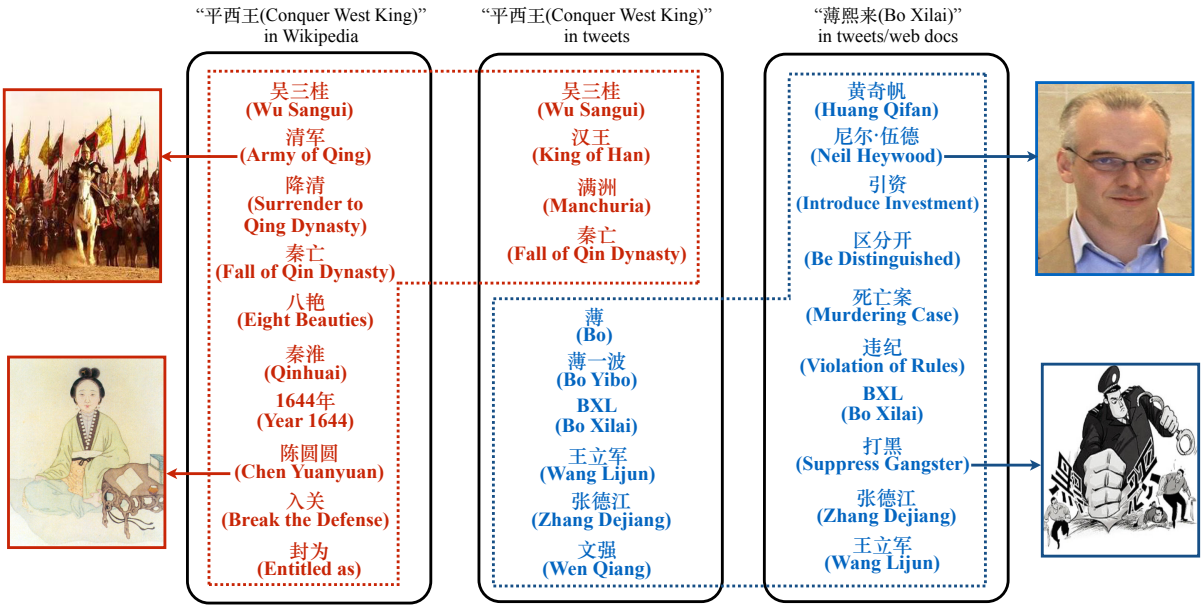

Figure 5: Top 10 closest entities to morph and target in different genres

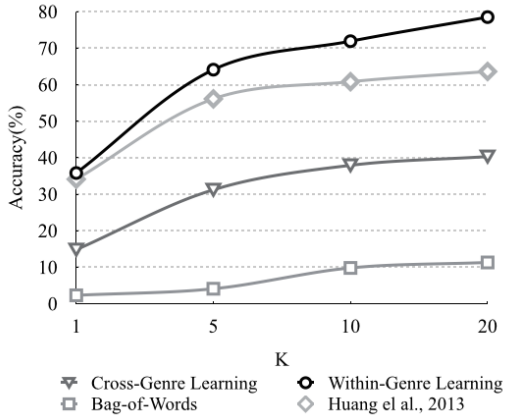

Figure 4: Resolution Acc@K for Perfect Morph Mentions

NLP tasks: entity mention extraction (e.g., (Zitouni and Florian, 2008; Ohta et al., 2012; Li and Ji, 2014)), metaphor detection (e.g., (Wang et al., 2006; Tsvetkov, 2013; Heintz et al., 2013)), word sense disambiguation (WSD) (e.g., (Yarowsky, 1995; Mihalcea, 2007; Navigli, 2009)), and entity linking (EL) (e.g., (Mihalcea and Csomai, 2007; Ji et al., 2010; Ji et al., 2011; Ji et al., 2014). However, none of these previous techniques can be applied directly to tackle this problem. As mentioned in section 3.1, entity morphs are fundamentally different from regular entity mentions. Our task is also different from metaphor detection because morphs cover a much wider range of semantic categories and can include either abstractive or concrete information. Some common features for detecting metaphors (e.g. (Tsvetkov, 2013)) are not effective for morph extraction: (1). Semantic categories. Metaphors usually fall into certain semantic categories such as noun.animal and noun.cognition. (2). Degree of abstractness. If the subject or an object of a concrete verb is abstract then the verb is likely to be a metaphor. In contrast, morphs can be very abstract (e.g., "函数 *(Function)*" refers to "杨幂 *(Yang Mi)*" because her first name "幂 *(Mi)*" means the Power Function) or very concrete (e.g., "薄督 *(Governor Bo)*" refers to "薄熙来 *(Bo Xilai)*"). In contrast to traditional WSD where the senses of a word are usually quite stable, the "sense" (target entity) of a morph may be newly emergent or evolve over time rapidly. The same morph can also have multiple senses. The EL task focuses more on explicit and formal entities (e.g., named entities), while morphs tend to be informal and convey implicit information.

Morph mention detection is also related to malware detection (e.g., (Firdausi et al., 2010; Chandola et al., 2009; Firdausi et al., 2010; Christodorescu and Jha, 2003)) which discovers abnormal behavior in code and malicious software. In contrast our task tackles anomaly texts in semantic context.

Deep learning-based approaches have been demonstrated to be effective in disambiguation related tasks such as WSD (Bordes et al., 2012), entity linking (He et al., 2013) and question linking (Yih et al., 2014; Bordes et al., 2014; Yang et al., 2014). In this paper we proved that it's cru-

cial to keep the genres consistent between learning embeddings and applying embeddings.

7 Conclusions and Future Work

This paper describes the first work of context-aware end-to-end morph decoding. By conducting deep analysis to identity the common characteristics of morphs and the unique challenges of this task, we leverage a large amount of unlabeled data and the coreferential and correlation relations to perform collective inference to extract morph mentions. Then we explore deep learning-based techniques to capture the semantics of morph mentions and entities and resolve morph mentions on the fly. Our future work includes exploiting the profiles of target entities as feedback to refine the results of morph mention extraction. We will also extend the framework for event morph decoding.

Acknowledgments

This work was supported by the US ARL NS-CTA No. W911NF-09-2-0053, DARPA DEFT No. FA8750-13-2-0041, NSF Awards IIS-1523198, IIS-1017362, IIS-1320617, IIS-1354329 and HDTRA1-10-1-0120, gift awards from IBM, Google, Disney and Bosch. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, March.

A. Bordes, X. Glorot, J. Weston, and Y. Bengio. 2012. Joint learning of words and meaning representations for open-text semantic parsing. In *Proc. of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS2012)*.

A. Bordes, S. Chopra, and J. Weston. 2014. Question answering with subgraph embeddings. In *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP2014)*.

V. Chandola, A. Banerjee, and V. Kumar. 2009. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):15.

P. Chang, M. Galley, and D. Manning. 2008. Optimizing chinese word segmentation for machine translation performance. In *Proc. of the Third Workshop on Statistical Machine Translation (StatMT 2008)*.

J. Chen, D. Ji, C Tan, and Z. Niu. 2006. Relation extraction using label propagation based semi-supervised learning. In *Proc. of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (ACL2006)*.

M. Christodorescu and S. Jha. 2003. Static analysis of executables to detect malicious patterns. In *Proc. of the 12th Conference on USENIX Security Symposium (SSYM2003)*.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, November.

C. Cortes and V. Vapnik. 1995. Support-vector networks. *Machine Learning*, 20:273–297, September.

I. Firdausi, C. Lim, A. Erwin, and A. Nugroho. 2010. Analysis of machine learning techniques used in behavior-based malware detection. In *Proc. of the 2010 Second International Conference on Advances in Computing, Control, and Telecommunication Technologies (ACT2010)*.

Z. Harris. 1954. Distributional structure. *Word*, 10:146–162.

Z. He, S. Liu, M. Li, M. Zhou, L. Zhang, and H. Wang. 2013. Learning entity representation for entity disambiguation. In *Proc. of the 51st Annual Meeting of the Association for Computational Linguistics (ACL2013)*.

I. Heintz, R. Gabbard, M. Srivastava, D. Barner, D. Black, M. Friedman, and R. Weischedel. 2013. Automatic extraction of linguistic metaphors with lda topic modeling. In *Proc. of the ACl2013 Workshop on Metaphor in NLP*.

H. Huang, Z. Wen, D. Yu, H. Ji, Y. Sun, J. Han, and H. Li. 2013. Resolving entity morphs in censored data. In *Proc. of the 51st Annual Meeting of the Association for Computational Linguistics (ACL2013)*.

H. Huang, Y. Cao, X. Huang, H. Ji, and C. Lin. 2014. Collective tweet wikification based on semi-supervised graph regularization. In *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL2014)*.

H. Ji, R. Grishman, H.T. Dang, K. Griffitt, and J. Ellis. 2010. Overview of the tac 2010 knowledge base population track. In *Proc. of the Text Analysis Conference (TAC2010)*.

H. Ji, R. Grishman, and H.T. Dang. 2011. Overview of the tac 2011 knowledge base population track. In *Proc. of the Text Analysis Conference (TAC2011)*.

H. Ji, J. Nothman, and H. Ben. 2014. Overview of tac-kbp2014 entity discovery and linking tasks. In *Proc. of the Text Analysis Conference (TAC2014)*.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of the Eighteenth International Conference on Machine Learning (ICML2001)*.

Q. Li and H. Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL2014)*.

Q. Li, H. Li, H. Ji, W. Wang, J. Zheng, and F. Huang. 2012. Joint bilingual name tagging for parallel corpora. In *Proc. of the 21st ACM International Conference on Information and Knowledge Management (CIKM2012)*.

R. Mihalcea and A. Csomai. 2007. Wikify!: linking documents to encyclopedic knowledge. In *Proc. of the sixteenth ACM conference on Conference on information and knowledge management (CIKM2007)*.

R. Mihalcea. 2007. Using wikipedia for automatic word sense disambiguation. In *Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL2007)*.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

T. Mikolov, I. Sutskever, K. Chen, S.G. Corrado, and J. Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*.

R. Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41:10:1–10:69, February.

Z. Niu, D. Ji, and C. Tan. 2005. Word sense disambiguation using label propagation based semi-supervised learning. In *Proc. of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL2005)*.

T. Ohta, S. Pyysalo, J. Tsujii, and S. Ananiadou. 2012. Open-domain anatomical entity mention detection. In *Proc. of the ACL2012 Workshop on Detecting Structure in Scholarly Discourse*.

A. Smola and R. Kondor. 2003. Kernels and regularization on graphs. In *Proc. of the Annual Conference on Computational Learning Theory and Kernel Workshop (COLT2003)*.

K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL2003)*.

Y. Tsvetkov. 2013. Cross-lingual metaphor detection using common semantic features. In *Proc. of the ACL2013 Workshop on Metaphor in NLP*.

Z. Wang, H. Wang, H. Duan, S. Han, and S. Yu. 2006. Chinese noun phrase metaphor recognition with maximum entropy approach. In *Proc. of the Seventh International Conference on Intelligent Text Processing and Computational Linguistics (CICLing2006)*.

M. Yang, N. Duan, M. Zhou, and H. Rim. 2014. Joint relational embeddings for knowledge-based question answering. In *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP2014)*.

D. Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proc. of the 33rd Annual Meeting on Association for Computational Linguistics (ACL1995)*.

W. Yih, X. He, and C. Meek. 2014. Semantic parsing for single-relation question answering. In *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL2014)*.

H. Zhang, H. Yu, D. Xiong, and Q. Liu. 2003. Hhmm-based chinese lexical analyzer ictclas. In *Proc. of the second SIGHAN workshop on Chinese language processing (SIGHAN2003)*.

B. Zhang, H. Huang, X. Pan, H. Ji, K. Knight, Z. Wen, Y. Sun, J. Han, and B. Yener. 2014. Be appropriate and funny: Automatic entity morph encoding. In *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL2014)*.

D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. 2004. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, pages 321–328.

X. Zhu, Z. Ghahramani, and J. Lafferty. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *Proc. of the International Conference on Machine Learning (ICML2003)*.

I. Zitouni and R. Florian. 2008. Mention detection crossing the language barrier. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP2008)*.