"A Spouse Begins with a Deletion of *engage* and Ends with an Addition of *divorce*": Mapping Text and Infobox Changes in Wikipedia to Learn Verbs and State Changes for Knowledge Base Updates

Abstract

Most knowledge bases (KBs) in recent years are static. They contain facts about the world yet are seldom updated as the world changes. This paper proposes a method for learning state changes brought about by verbs on its arguments (i.e., entities). State changes are viewed as updates of KB facts pertaining to the entities. We propose to learn state changes brought about by verbs using Wikipedia revision histories. When a state-changing event happens to an entity, the Wikipedia infobox that contains facts of the entity may be updated. At the same time, text that contain verbs that express the event may also be added to/deleted from the entity's Wikipedia page. We use Wikipedia revisions as distantly supervised data to automatically learn verbs and state changes. We also use constraints such as mutually exclusive changes vs. simultaneous changes of infobox slots to effectively map verbs to infobox changes. We observe in our experiments that when state-changing verbs are being added to/deleted from a person's Wikipedia text, we can update infobox facts about the person effectively (with an 89% precision and 74% recall).

1 Introduction

In recent years there has been a lot of research on extracting relational facts between entities and storing them in knowledge bases (KBs). These knowledge bases such as YAGO (which extract facts from Wikipedia infoboxes (Suchanek et al., 2007)) or NELL (which extracts facts from any Web text (Carlson et al., 2010; Fader et al., 2011)) are generally static. They are not updated as the Web changes when in reality new facts arise while others cease to be valid. One approach towards real-time population of KBs is to extract facts from dynamic content of the web such as news (Nakashole and Weikum, 2012). This paper proposes a *shift* of focus from doing KB updates by extracting facts in text to doing them by identifying state changes brought about by verbs in text.

The benefit of such shift is multi-fold: (1) In relation extraction, both *marry* and *divorce* are good patterns for extracting the SPOUSE relation. But by identifying that they bring about different state changes: *marry* signals the start while *divorce* signals the end of the SPOUSE relation; we can update the entity's fact *and* its temporal scope (Wijaya et al., 2014). (2) Learning state changes brought about by verbs can pave ways to learning the pre- and post-conditions of state-changing verbs: the entry condition (in terms of KB facts) that must be true for an event expressed by the verb to take place, and the exit condition (in terms of KB facts) that will be true after the event. Such pre- and post-conditions can be useful for (a) learning event sequences as a collection of verbs chained together by pre- and post-condition of their shared entities, (b) for inferring cascading effect of an event via the pre- and post-condition of shared entities in an event sequence, or (c) for inferring unknown states of entities from the verbs they participate in.

In this paper, we propose to learn state changes brought about by verbs using Wikipedia revision histories. Our assumption is that when a state-changing event happens to an entity e.g., a marriage, its Wikipedia infobox: a structured document that contains a set of facts (attribute-value pairs) of the entity is updated e.g., by the addition of a new SPOUSE value. At the same time, texts that contain verbs that express the event e.g., *wed* may be added to the entity's Wikipedia page (see an example in Figure 1). Wikipedia revisions

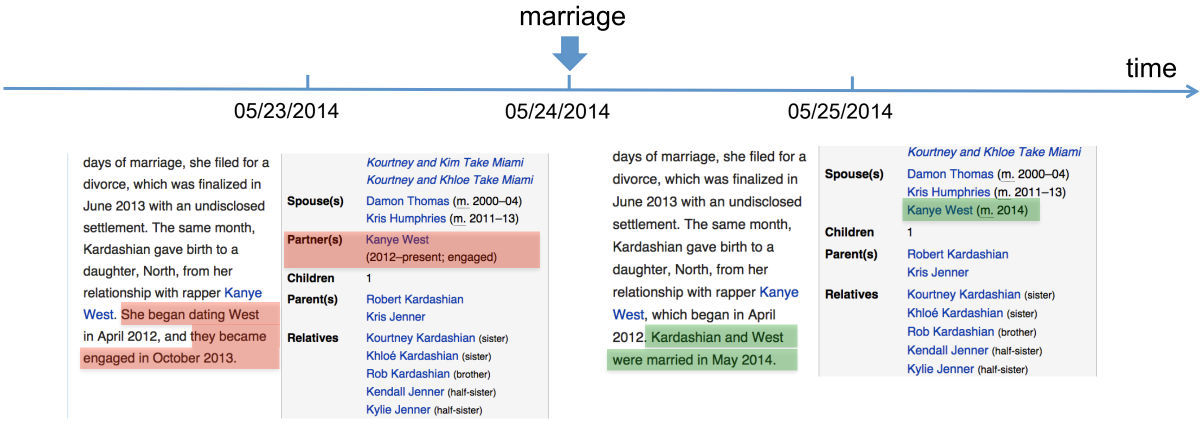

Figure 1: A snapshot of Kim Kardashian's Wikipedia revision history, highlighting text and infobox changes. In red (and green) are the difference between the page on 05/25/2014 and 05/23/2014: things that are deleted from (and resp. added to) the page.

over many entities can act as distantly supervised data for mapping text and infobox changes that relate to events. However, Wikipedia revisions are *noisy*: there is no guarantee that only the infobox slots related to a particular event will be updated. For example, when an event such as death happens, slots regarding birth e.g., *birthdate*, *birthplace*, may also be updated. To alleviate the effect of such, we leverage *typical* constraints between state changes e.g., that the start of *deathdate* is mutually exclusive i.e., cannot happen at the same time as the start of *birthdate* or that the start of *birthdate* is simultaneous with the start of *birthplace*, to effectively learn infobox changes that relate to a particular event-expressing verb.

Our contribution is (1) the construction and use of an interesting, distantly labeled, dataset from Wikipedia revisions to learn about verbs and state changes, and (2) the learned resource of verbs that is effective for identifying state changes[1].

2 Method

2.1 Data Construction

We construct a dataset from Wikipedia revision histories of person entities whose facts change between the year 2007 and 2012 (i.e., have at least one fact in YAGO KB with a start or end time in this period). We obtain Wikipedia URLs of this set of entities P from YAGO and crawl their revision histories. Given a person p, his Wikipedia revision history H_p has a set of ordered dates T_p on which revisions are made to his Wikipedia page W_p (we consider a date granularity for time). Each revision $W_{p,t_p} \in H_p$ is the content of W_p at date t_p where $t_p \in T_p$.

A document d_{p,t_p} in our data set is the *difference*[2] between any two consecutive revisions to W_p that is separated by at least a single date worth of revisions i.e., $d_{p,t_p} = W_{p,\overline{t_p+2}} - W_{p,\underline{t_p}}$. Where $W_{p,\overline{t_p+2}}$ is the *first* revision on date $t_p + 2$ and $W_{p,\underline{t_p}}$ is the *last* revision on date t_p (since W_p can be revised multiple times on a date). Our dataset consists of all documents d_{p,t_p}, $\forall t_p \in T_p$, $t \in [01/01/2007, 12/31/2012]$, and $\forall p \in P$; a total of 288,184 documents from revision histories of 16,909 Wikipedia entities.

Each Wikipedia revision W_{p,t_p} consists of a set of infobox slots S and a textual content C, where each slot $s \in S$ is a quadruple, $\langle s_{att}, s_{value}, s_{start}, s_{end} \rangle$ containing the attribute name (non-empty), the attribute value, and the start and end time for which this attribute-value pair is valid.

Each document in our dataset is a *difference* between W_{p,t_p+2} and W_{p,t_p}, and therefore consists of a set of infobox changes ΔS and textual changes ΔC. Each slot change $\delta s \in \Delta S$ is also a quadruple but has s_{value}, s_{start}, or s_{end}, prefixed with $+$ or $-$ to indicate whether they are added or deleted in W_{p,t_p+2}. Similarly, each text change $\delta c \in \Delta C$ is prefixed with $+$ or $-$ to indicate whether they

[1] We make our dataset and verbs resource available here: http://.../verbs.html

[2] a HTML document obtained by "compare selected revisions" functionality in Wikipedia

are added or deleted in W_{p,t_p+2}. For example in Figure 1, a document $d_{kim,\ 05/23/2014} = W_{kim,05/25/2014} - W_{kim,05/23/2014}$ consists of slot changes: ⟨SPOUSE, +"Kanye West", +"2014", " "⟩, ⟨PARTNER, −"Kanye West", −"2012-present; engaged", " "⟩ and text changes: +"Kardashian and West were married in May 2014", −"She began dating West", −"they became engaged in October 2013".

For each document, we use its slot changes ΔS to label the document and its text changes ΔC as features for the document. We label documents that have $\langle s_{att}, +s_{value}, *, * \rangle \in \Delta S$ or $\langle s_{att}, *, +s_{start}, * \rangle \in \Delta S$ with the label *begin-*s_{att} and documents that have $\langle s_{att}, *, *, +s_{end} \rangle \in \Delta S$ with the label *end-*s_{att}. The label represents the state change that happens in the document. For example, in Figure 1, $d_{kim,\ 05/23/2014}$ is labeled with *begin-spouse* and *end-partner*.

As features, for each labeled document d_{p,t_p}, we extract verbs (and verbs+prepositions) in ΔC whose subject (*arg1*) matched its entity p and whose object (*arg2*) matched any of its δs value (or vice versa). We use 90% of our labeled documents as training and test on the remaining 10%. The task is to predict for each document, the label of the document given its verbs features.

2.2 Model

We use a Maximum Entropy (MAXENT) classifier given the set of training data = $\{(\mathbf{v}_{d_\ell},\ \mathrm{y})\}$ where $\mathbf{v}_{d_\ell} = (v_1,\ v_2,\ ...\ v_{|V|}) \in R^{|V|}$ is the $|V|$-dimensional representation of a labeled document d_ℓ i.e., counts of the verbs in the document, and y is the document's label as defined in section 2.1.

These training documents are used to estimate a set of weight vectors $\{\mathbf{w}_1,\ \mathbf{w}_2,\ ...\ \mathbf{w}_{|Y|}\}$, one for each label y. The classifier can then be applied to classify an unlabeled document d_u using:

$$p(y|\mathbf{v}_{d_u}) = \frac{\exp(\mathbf{w}_y \cdot \mathbf{v}_{d_u})}{\sum_{y'} \exp(\mathbf{w}_{y'} \cdot \mathbf{v}_{d_u})} \tag{1}$$

2.3 Feature Selection using Constraints

While feature weights obtained by MAXENT allow us to identify verbs that are good features for predicting a particular state change label, our distantly supervised training data is inherently noisy. For example, when an entity's death happens, birth-related information in the infobox may also be updated. This can lead to incorrect state change prediction. While using many distantly labeled data may help, improvements may be possible by leveraging constraints among infobox changes to select consistent verb features for each change.

The constraints that we use can be categorized into: (1) mutual exclusion (mutex) constraints which indicate that mutex changes should not *typically* happen at the same time e.g., the start of *deathdate* is mutex with all other non-death changes. A good *base* verb for one slot change e.g., "marry" for *begin-spouse* is therefore not a good feature for other, mutex slot change *end-spouse*. (2) Simultaneous constraints which indicate that simultaneous changes should *typically* happen at the same time e.g., the start of *birthdate* is simultaneous with the start of other birth-related changes such as *birthplace*, *birthname*, *birthcity*. A good base verb for one slot change e.g., "die" for *deathdate* is therefore also a good feature for other, simultaneous, slot changes: *deathplace*, *deathcause*, etc. We compile such constraints using heuristics on the label names.

Given the

3 Experiments

4 Related Works

Related works here

5 Conclusion

Acknowledgments

We thank members of the NELL team at CMU for their helpful comments. This research was supported by DARPA under contract number FA8750-13-2-0005.

References

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. 2010. Toward an architecture for never-ending language learning. In *AAAI*, volume 5, page 3.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics.

Ndapandula Nakashole and Gerhard Weikum. 2012. Real-time population of knowledge bases: opportunities and challenges. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 41–45. Association for Computational Linguistics.

Roger C Schank and Robert P Abelson. 2013. *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*. Psychology Press.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM.

Yafang Wang, Mingjie Zhu, Lizhen Qu, Marc Spaniol, and Gerhard Weikum. 2010. Timely yago: harvesting, querying, and visualizing temporal knowledge from wikipedia. In *Proceedings of the 13th International Conference on Extending Database Technology*, pages 697–700. ACM.

Derry Wijaya, Ndapa Nakashole, and Tom Mitchell. 2014. Ctps: Contextual temporal profiles for time scoping facts via entity state change detection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.