"A Spousal Relation Begins with a Deletion of *engage* and Ends with an Addition of *divorce*": Learning State Changing Verbs from Wikipedia Revision History

Abstract

Learning to determine when the facts of a Knowledge Base (KB) have to be updated is a challenging task. We propose to learn state changing verbs from Wikipedia edit history. When a state-changing event, such as a marriage or death, happens to an entity, the infobox on the entity's Wikipedia page usually gets updated. At the same time, the article text may be updated with verbs either being added or deleted to reflect the changes made to the infobox. We use Wikipedia edit history to distantly supervise a method for automatically learning verbs and state changes. Additionally, our method uses constraints to effectively map verbs to infobox changes. We observe in our experiments that when state-changing verbs are added or deleted from an entity's Wikipedia page text, we can update the entity's infobox updates with 88% precision and 76% recall. One compelling application of our verbs is to incorporate them as triggers in methods for updating existing KBs, which are currently mostly static.

1 Introduction

Extracting relational facts between entities and storing them in knowledge bases (KBs) has been a topic of active research in recent years . The resulting KBs are generally static and are not updated as the facts change. (Suchanek et al., 2007; Carlson et al., 2010; Fader et al., 2011; Mitchell et al., 2015) One possible approach to updating KBs is to extract facts from dynamic Web content such as news (Nakashole and Weikum, 2012). In this paper, we propose to predict state changes caused by verbs acting on entities in text. This is different from simply applying the same text extraction pipeline, that created the original KB, to dynamic Web content.

In particular, our approach has the following advantages: (1) Consider for example the SPOUSE relation, both *marry* and *divorce* are good patterns for extracting this relation. In our work, we wish to learn that they cause different state changes. Thus, we can update the entity's fact *and* its temporal scope (Wijaya et al., 2014a). (2) Learning state changing verbs can pave ways for learning the ordering of verbs in terms of their pre- and post-conditions.

Our approach learns state changing verbs from Wikipedia revision history. In particular, we seek to establish a correspondence between infobox edits and verbs edits in the same article. The infobox of a Wikipedia article is a structured box that summarizes an entity as a set of facts (attribute-value pairs) . Our assumption is that when a state-changing event happens to an entity e.g., a marriage, its Wikipedia infobox is updated by adding a new SPOUSE value. At the same time, the article text might be updated with verbs that express the event, e.g., *X is now **married** to Y*. Figure 1 is an example of an infobox of entity changing at the same time as the article's main text to reflect an marriage event.

Wikipedia revision history of many articles can act as distant supervision data for learning the correspondence between text and infobox changes. However, these revisions are very *noisy*. Many infobox slots can be updated when a single event happens. For example, when a death happens, slots regarding birth e.g., *birthdate*, *birthplace*, may also be updated or added if they were missing before. Therefore, our method has to handle these sources of noise. We leverage logical constraints to rule out meaningless mappings between infobox and text changes.

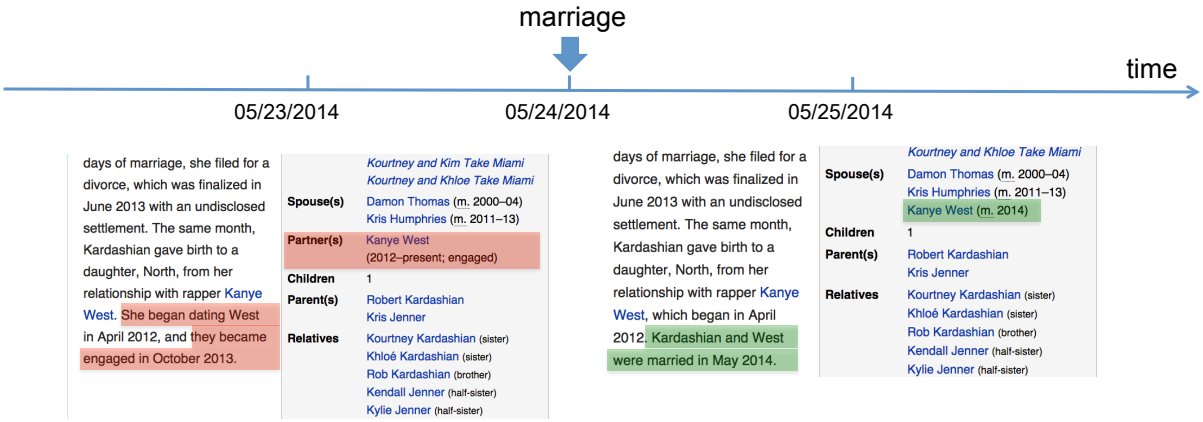

Figure 1: A snapshot of Kim Kardashian's Wikipedia revision history, highlighting text and infobox changes. In red (and green) are the differences between the page on 05/25/2014 and 05/23/2014: things that are deleted from (and added to) the page.

In summary, our contributions are as follows: (1) We extracted a distantly labeled dataset from Wikipedia revision history targeted for the task of learning verbs that cause infobox updates, which we showed to be effective for learning state changing verbs. (2) We developed a method for learning state changing verbs from Wikipedia revision history. (3) A resource that contains state changing verbs, which we make available for future research [1]. We also make the revision history dataset available.

2 Method

2.1 Data Construction

We construct a dataset from Wikipedia edit history of person entities whose facts change between the year 2007 and 2012 (i.e., have at least one fact in YAGO KB (Suchanek et al., 2007) with a start or end time in this period). We obtain Wikipedia URLs of this set of entities P from YAGO and crawl their article's revision history. Given a person p, his Wikipedia revision history R_p has a set of ordered dates T_p on which revisions are made to his Wikipedia page (we consider date granularity). Each revision $r_{p,t_p} \in R_p$ is his Wikipedia page 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 separated by at least a single date i.e., $d_{p,t_p} = r_{p,t_p+2} - r_{p,t_p}$. Where r_{p,t_p+2} is the *first* revision on date $t_p + 2$ and r_{p,t_p} is the *last* revision on date t_p (since a page can be revised many times in a day). Our dataset consists of all documents d_{p,t_p}, $\forall t_p \in T_p,\ t_p \in [01/01/2007,\ 12/31/2012]$, and $\forall p \in P$; a total of 288,184 documents from revision histories of 16,909 Wikipedia entities.

Each Wikipedia revision r_{p,t_p} is a set of infobox slots S_{p,t_p} and textual content C_{p,t_p}, where each slot $s \in S_{p,t_p}$ 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 r_{p,t_p+2} and r_{p,t_p}, and is a set of infobox changes $\Delta S_{p,t_p}$ and textual changes $\Delta C_{p,t_p}$. Each slot change $\delta s \in \Delta S_{p,t_p} = \langle s_{att}, \delta s_{value}, \delta s_{start}, \delta s_{end} \rangle$, where δs_{value}, δs_{start}, or δs_{end}, whenever not empty, is prefixed with $+$ or $-$ to indicate whether they are added or deleted in r_{p,t_p+2}. Similarly, each text change $\delta c \in \Delta C_{p,t_p}$ is prefixed with $+$ or $-$ to indicate whether they are added or deleted in r_{p,t_p+2}. For example, in Figure 1, a document $d_{kim,\ 05/23/2014} = r_{kim,05/25/2014} - r_{kim,05/23/2014}$ is a set of slot changes: $\langle$SPOUSE, +"Kanye West", +"2014", " "$\rangle$, $\langle$PARTNER, −"Kanye West", −"2012-present; engaged", " "$\rangle$ and a set of text changes: +"Kardashian and West were married in May 2014", −"She began dating West", −"they became engaged in October 2013".

For each d_{p,t_p}, we use $\Delta S_{p,t_p}$ to label the document and $\Delta C_{p,t_p}$ to extract features for the document. We label d_{p,t_p}

[1] URL retracted for blind reviews

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

that has $\langle s_{att}, +\delta s_{value}, *, * \rangle \in \Delta S_{p,t_p}$ or $\langle s_{att}, *, +\delta s_{start}, * \rangle \in \Delta S_{p,t_p}$ with the label *begin-*s_{att} and d_{p,t_p} that has $\langle s_{att}, *, *, +\delta s_{end} \rangle \in \Delta S_{p,t_p}$ with the label *end-*s_{att}. The label represents the state change that happens in d_{p,t_p}. For example, in Figure 1, $d_{kim,\ 05/23/2014}$ is labeled with *begin-spouse* and *end-partner*. As features, for each labeled d_{p,t_p}, we extract verbs (or verbs+prepositions) $v \in \Delta C_{p,t_p}$ that have ($v_{subject}$, v_{object}) = ($_{arg1}$, $_{arg2}$) or ($v_{subject}$, v_{object}) = ($_{arg2}$, $_{arg1}$), where $_{arg1}$= p and $\langle s_{att},_{arg2}, *, * \rangle$ or $\langle s_{att}, *,_{arg2}, * \rangle$ or $\langle s_{att}, *, *,_{arg2} \rangle$ is $\in \Delta S_{p,t_p}$.

2.2 Model

We use a Maximum Entropy (MAXENT) classifier given a set of training data = $\{(\mathbf{v}_{d_\ell},\ y)\}$ 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 where V is the set of all verbs in our training data, and y is the label of d_ℓ as defined in 2.1.

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

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

2.3 Feature Selection using Constraints

While feature weights from the MAXENT model allow us to identify verbs that are good features for predicting a particular state change label, our distantly supervised training data is inherently noisy. Changes to multiple infoboxes can happen within our revision. We therefore utilize constraints among state changes to select consistent verb features for each type of state change.

We use two types of constraints: (1) mutual exclusion (*Mutex*) which indicate that mutex state changes do not happen at the same time e.g., update on *birthdate* should not *typically* happen with update on *deathcause*. Hence, their state changing verbs should be different. (2) Simultaneous (*Sim*) constraints which indicate that simultaneous state changes should *typically* happen at the same time e.g., update on *birthdate* should *typically* happen with other birth-related updates. We manually specified these two types of constraints to all pairs infoboxes where they apply..

Given a set of constraints, a set of labels Y, and a set of base verbs[3] B in our training data, we solve a Mixed-Integer Program (MIP) for each base verb $b \in B$ to estimate whether b should be a feature for state change $y \in Y$.

We obtain label membership probabilities $\{P(y|b) = count(y,b)/\sum_{y'} count(y',b)\}$ from our training data. The MIP takes the scores $P(y|b)$ and constraints as input and produces a bit vector of labels $\mathbf{a}_b$ as output, each bit $a_b^y \in \{0,\ 1\}$ representing whether b should be a feature for y/not.

The MIP formulation for a base verb b is presented by Equation 2. For each b, this method tries to maximize the sum of scores of selected labels, after penalizing for violation of label constraints. Let $\zeta_{y,y'}$ be slack variables for *Sim* constraints, and $\xi_{y,y'}$ be slack variables for *Mutex* constraints.

$$\begin{aligned} \underset{\mathbf{a}_b,\ \zeta_{y,y'},\ \xi_{y,y'}}{\text{maximize}} \quad & \Big(\sum_y a_b^y * P(y|b) - \sum_{\langle y,y' \rangle \in Sim} \zeta_{y,y'} - \sum_{\langle y,y' \rangle \in Mutex} \xi_{y,y'} \Big) \\ \text{subject to} \quad & (a_b^y - a_b^{y'})^2 \leq \zeta_{y,y'}, \quad \forall \langle y,y' \rangle \in Sim \\ & a_b^y + a_b^{y'} \leq 1 + \xi_{y,y'}, \quad \forall \langle y,y' \rangle \in Mutex \\ & \zeta_{y,y'},\ \xi_{y,y'} \geq 0,\ a_b^y \in \{0,1\}, \quad \forall y,y' \end{aligned} \quad (2)$$

Solving MIP per base verb is fast; we reduce the number of labels considered per base verb i.e., we only consider a label y to be a candidate for b if $\exists\ v_i \in V$ s.t. $w_y^i > 0$ and b = base form of v_i (after removing preposition). Then, we need to only solve MIP for base verbs that have non-empty candidate labels.

After we output $\mathbf{a}_b$ for each base verb, we do feature selection on the learned verb features of each label. We only select a verb v_i to be a feature for y if the learned weight $w_y^i > 0$ and $a_b^y = 1$, where b = the base form of v_i. Essentially for each label, we are choosing verb features that have positive weights and are consistent for the label.

3 Experiments

We use 90% of our labeled documents as train and test on the remaining 10%. Since revision history data is noisy, we manually go through our test data to discard documents that have incorrect infobox labels by looking the text that changed. The task is to predict for each document (revision), the label (infobox slot value) of the document given its verbs features. We compute precision, recall, and F1 values of our predictions and compare the values before and after feature selection (Figure 2).

[3]The verb root or base form of a verb

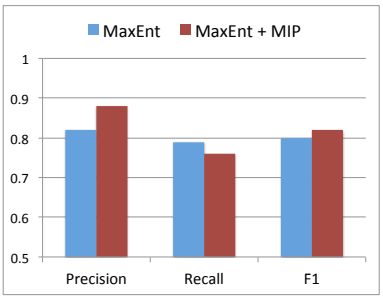

Figure 2: Results of predicting state change labels (infobox types) using verb features.

Label	Verb
begin-deathdate	+(arg1) die on (arg2), +(arg1) die (arg2), +(arg1) pass on (arg2)
begin-birthplace	+(arg1) be born in (arg2), +(arg1) bear in (arg2), +(arg1) be born at (arg2)
begin-predecessor	+(arg1) succeed (arg2), +(arg1) replace (arg2), +(arg1) join cabinet as (arg2), +(arg1) join as (arg2)
begin-successor	+(arg1) lose **seat** to (arg2), +(arg1) resign on (arg2), +(arg1) resign from post on (arg2)
begin-termstart	+(arg1) be appointed on (arg2), +(arg1) serve from (arg2), +(arg1) be elected on (arg2)
begin-spouse	+(arg1) marry on (arg2), +(arg1) marry (arg2), +(arg1) be married on (arg2), -(arg1) be engaged to (arg2)
end-spouse	+(arg1) file **for divorce** in (arg2), +(arg1) die on (arg2), +(arg1) divorce in (arg2)
begin-almamater	+(arg1) graduate from (arg2), +(arg1) attend (arg2), +(arg1) be educated at (arg2)
begin-youthclubs	+(arg1) start career with (arg2), +(arg1) begin **career** with (arg2), +(arg1) start with (arg2)

Table 1: Comparison of verb phrases learned before and after feature selection for various labels (infobox types).

We observe the value of doing feature selection by asserting constraints in an MIP formulation in Figure 2. Feature selection improves precision without significantly reducing recall; resulting in a better F1. By asserting constraints, some of the inconsistent verb features for the labels were removed. For example, before feature selection, the verbs: "marry", "marry in" and "be married to" were high-weighted features for both *begin-spouse* and *end-spouse*. After asserting constraints that *begin-spouse* is mutex with *end-spouse*, these verbs (whose base form is "marry") are filtered out from the features of *end-spouse*. We show some of the learned verb features (after feature selection) for some of the labels in (Table 1).

4 Related Work

Learning from Wikipedia Revision History. Wikipedia edit history has been exploited in a number of problems. A popular task in this regard is that of Wikipedia edit history categorization(Daxenberger and Gurevych, 2013). This task involves characterizing a given edit instance as one of many possible categories such as spelling error correction, paraphrasing, vandalism, and textual entailment. (Nelken and Yamangil, 2008; Cahill et al., 2013; Zanzotto and Pennacchiotti, 2010; Recasens et al., 2013).. Prior methods target various tasks different from ours.

Learning State Changing Verbs. Very few works have studied the problem of learning state changing verbs. (Hosseini et al., 2014) learned state changing verbs in the context of solving arithmetic word problems. They learned the effect of words such as *add, subtract* on the current state. The VerbOcean resource was automatically generated from the Web(Chklovski and Pantel, 2004). The authors studied the problem of fine-grained semantic relationships between verbs. They learn relations such as if someone has bought an item, they may sell it at a later time. This then involves capturing empirical regularities such as "X buys Y" happens before "X sells Y". Unlike the work we present here, the methods of (Chklovski and Pantel, 2004; Hosseini et al., 2014) do not make a connection to knowledge base relations such as Wikipedia infoboxes. In a vision paper, (Wijaya et al., 2014b) give high level descriptions of a number of possible methods for learning state changing methods. They did not implement any of them.

5 Conclusion

In this paper we presented a method that learns state changing verb phrases from Wikipedia revision history. We first constructed and curated a novel dataset from Wikipedia revision history that is tailored to our task. We showed that this dataset is useful for learning verb phrase features that are effective for predicting state changes in the knowledge base (KB), where we considered the KB to be infoboxes and their values. As future work we wish to explore the usefulness of our verb resource to other KBs in order to improve KB freshness. This is important because most existing KBs are mostly static.

References

Aoife Cahill, Nitin Madnani, Joel Tetreault, and Diane Napolitano. 2013. Robust systems for preposi-

tion error correction using wikipedia revisions. In *In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

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.

Timothy Chklovski and Patrick Pantel. 2004. Verbocean: Mining the web for fine-grained semantic verb relations. In *Proceedings of EMNLP 2004*, pages 33–40.

Johannes Daxenberger and Iryna Gurevych. 2012. A corpus-based study of edit categories in featured and non-featured wikipedia articles. In *COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 8-15 December 2012, Mumbai, India*, pages 711–726.

Johannes Daxenberger and Iryna Gurevych. 2013. Automatically classifying edit categories in wikipedia revisions. In *EMNLP*, pages 578–589.

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.

Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014. Learning to solve arithmetic word problems with verb categorization. In *EMNLP*, pages 523–533. ACL.

Tom M. Mitchell, William W. Cohen, Estevam R. Hruschka Jr., Partha Pratim Talukdar, Justin Betteridge, Andrew Carlson, Bhavana Dalvi Mishra, Matthew Gardner, Bryan Kisiel, Jayant Krishnamurthy, Ni Lao, Kathryn Mazaitis, Thahir Mohamed, Ndapandula Nakashole, Emmanouil Antonios Platanios, Alan Ritter, Mehdi Samadi, Burr Settles, Richard C. Wang, Derry Tanti Wijaya, Abhinav Gupta, Xinlei Chen, Abulhair Saparov, Malcolm Greaves, and Joel Welling. 2015. Never-ending learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 2302–2310.

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.

Rani Nelken and Elif Yamangil. 2008. Mining wikipedias article revision history for training computational linguistics algorithms.

Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. 2013. Linguistic models for analyzing and detecting biased language. In *ACL (1)*, pages 1650–1659.

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.

Derry Wijaya, Ndapa Nakashole, and Tom Mitchell. 2014a. 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.

Derry Tanti Wijaya, Ndapandula Nakashole, and Tom M Mitchell. 2014b. Mining and organizing a resource of state-changing verbs. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*.

Fabio Massimo Zanzotto and Marco Pennacchiotti. 2010. Expanding textual entailment corpora from wikipedia using co-training.