# “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 revision history. When a state-changing even, such as a marriage or death, happens to an entity, the infobox on the entity's Wikipedia page may be updated. At the same time, the text the same article may be updated with verbs either being added or deleted to reflect the new state of the real world entity. We use Wikipedia revision 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 with 88% precision and 76% recall. One immediate application of our verbs is to incorporate as triggers in methods for updating existing KBs, which are currently mostly static.


## 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., 2014a). (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 over many entities can act as distantly supervised data for mapping text and infobox changes that relate to events. However, Wikipedia revisions are

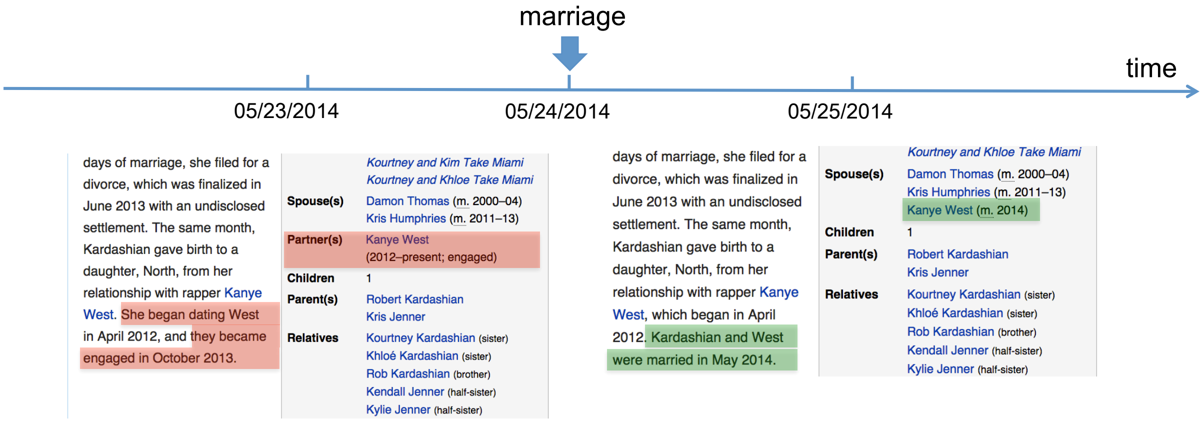


Figure 1: A snapshot of Kim Kardashian's Wikipedia revision history, highlighting text and infobox changes. In red (and green) are the difference between the page on 05/25/2014 and 05/23/2014: things that are deleted from (and resp. added to) the page.

*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 $R_p$ 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 $\delta s_{value}$, $\delta s_{start}$, or $\delta s_{end}$, 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, n 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: ⟨SPOUSE, +"Kanye West", +"2014", " "⟩, ⟨PARTNER, −"Kanye West", −"2012-present; engaged", " "⟩ and a set of text

[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

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 $d_{p,t_p}$ 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 the set of training data = $\{(\mathbf{v}_{d_\ell}, \text{ y})\}$ where $\mathbf{v}_{d_\ell} = (v_1, v_2, \ldots v_{|V|}) \in R^{|V|}$ is the $|V|$-dimensional representation of a labeled document $d_\ell$ where $V$ is the set of all verbs in our training data, and $y$ is the label of $d_\ell$ as defined in 2.1.

These training documents are used to estimate a set of weight vectors $\mathbf{w} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots \mathbf{w}_{|Y|}\}$, 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 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 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 state changes to select consistent verb features for each change.

The constraints that we use can be categorized into: (1) mutual exclusion (*Mutex*) which indicate that mutex state changes should not *typically* happen at the same time e.g., update on *birthdate* should not *typically* happen with update on *deathcause*. A good *base* verb[3] for one change e.g., "marry" for *begin-spouse* is therefore not a good feature for *end-spouse* (mutex with *begin-spouse*). (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. A good base verb for one state change e.g., "die" for *begin-deathdate* is therefore a good feature for *begin-deathdate*'s simultaneous changes: *begin-deathplace*, *begin-deathcause*, etc. We obtain such constraints using heuristics on our label names.

Given a set of constraints, a set of labels $Y$, and a set of base verbs $B$ in our training data, we solves 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 in 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.

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

Solving MIP per base verb is fast. To make it even more efficient, 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

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

| Method | Precision | Recall | F1 |
|---|---|---|---|
| MAXENT | 0.82 | 0.79 | 0.80 |
| MAXENT + MIP | 0.88 | 0.76 | 0.82 |

Table 1: Results of predicting state change label using verb features.

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 our data is noisy, we manually go through our test data to discard documents that have incorrect label to the change in its text. The task is to predict for each document, the label of the document given its verbs features. We compute precision, recall, and F1 values of our predictions of the test set and compare the values before and after feature selection (Table 1).

We observe the value of doing feature selection by asserting constraints in an MIP formulation in Table 1. Feature selection improves precision without reducing too much recall; thus resulting in a better F1. Some inconsistent verb features for the labels were removed by asserting constraints. 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 state change labels in (Table 2).

| Label | Verb |
|---|---|
| *begin-deathdate* | +(arg1) die on (arg2), +(arg1) die (arg2), +(arg1) pass on (arg2) |
| *begin-deathplace* | +(arg1) die in (arg2), +(arg1) die at (arg2), +(arg1) move to (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), +(arg1) lose election to (arg2) |
| *begin-occupation* | +(arg1) work as (arg2), +(arg1) nominate for (arg2), +(arg1) establish as (arg2) |
| *begin-termstart* | +(arg1) be appointed on (arg2), +(arg1) serve from (arg2), +(arg1) be elected on (arg2) |
| *begin-termend* | +(arg1) resign on (arg2), +(arg1) step down in (arg2), +(arg1) flee in (arg2) |
| *begin-office* | +(arg1) be appointed as (arg2), +(arg1) serve as (arg2), +(arg1) be appointed (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), +(arg1) announce **separation** on (arg2) |
| *begin-children* | +(arg1) have **child** (arg2), +(arg1) raise daughter (arg2), +(arg1) raise (arg2) |
| *begin-almamater* | +(arg1) graduate from (arg2), +(arg1) attend (arg2), +(arg1) be educated at (arg2) |
| *begin-awards* | +(arg1) be awarded (arg2), +(arg1) be named on (arg2), +(arg1) receive (arg2) |
| *begin-youthclubs* | +(arg1) start career with (arg2), +(arg1) begin **career** with (arg2), +(arg1) start with (arg2), +(arg1) play for (arg2) |
| *begin-clubs* | +(arg1) play for (arg2), +(arg1) play during career with (arg2), +(arg1) sign with (arg2), +(arg1) complete **move** to (arg2) |
| *begin-nationalteam* | +(arg1) make **appearance** for (arg2), +(arg1) make debut for (arg2), +(arg1) play for (arg2) |

Table 2: Comparison of verb features before and after feature selection. The texts in bold are (prep+) noun that occur most frequently with the combination of the (verb, label) in the train data.

## 4 Related Work

**Learning from Wikipedia Edit History.** Wikipedia edit history has been exploited in a number of language understanding problems. However, prior methods were targeted for various tasks different form ours. A popular task in this regard is that of Wikipedia edit history categorization(Daxenberger and Gurevych, 2013). This task involves characterizing a given change instance as one of many possible categories such as such as spelling error correction, paraphrase or vandalism to edits in a document. (Daxenberger and Gurevych, 2012) came up with a 21 category edit classification taxonomy. Other tasks to leverage Wikipedia edit history include: spelling error correction, summarization, preposition error correction, sentence compression, bias detection, and textual entailment (Nelken and Yamangil, 2008; Cahill et al., 2013; Zanzotto and Pennacchiotti, 2010; Recasens et al., 2013). These studies are concerned with coarse grained change type classification as opposed to the establishing a specific correspondence between text changes, at a word level, to infobox changes.

**Learning State Changing Verbs.** Very few works have studied the problem of detecting state changing verbs. (Hosseini et al., 2014) learned state changing verbs in the context of solving arithmetic word problems. The learned the effect of the words such as add, subtract on the 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 from Web document. 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 if “X buys Y” happens before “X sells Y”, for the same X and Y values in a given context. 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 but did not implement any of them.

## 5 Conclusion

In this paper we propose a method for learning about verbs and the state changes they bring about to KB facts. We construct a novel dataset from Wikipedia revision histories that is useful for learning about events, verbs and state changes. We observe in our experiments that learned verbs resource from this dataset is effective for predicting state changes in the knowledge base.


## 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


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

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.