

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

**Derry Tanti Wijaya**  
Carnegie Mellon University  
5000 Forbes Avenue  
Pittsburgh, PA, 15213  
dwijaya@cs.cmu.edu

**Ndapandula Nakashole**  
Carnegie Mellon University  
5000 Forbes Avenue  
Pittsburgh, PA, 15213  
ndapa@cs.cmu.edu

**Tom M. Mitchell**  
Carnegie Mellon University  
5000 Forbes Avenue  
Pittsburgh, PA, 15213  
tom.mitchell@cs.cmu.edu

## Abstract

Learning to determine when the time-varying 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 predict 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 the way 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 approximately 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 an entity changing at the same time as the article’s main text to reflect a 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

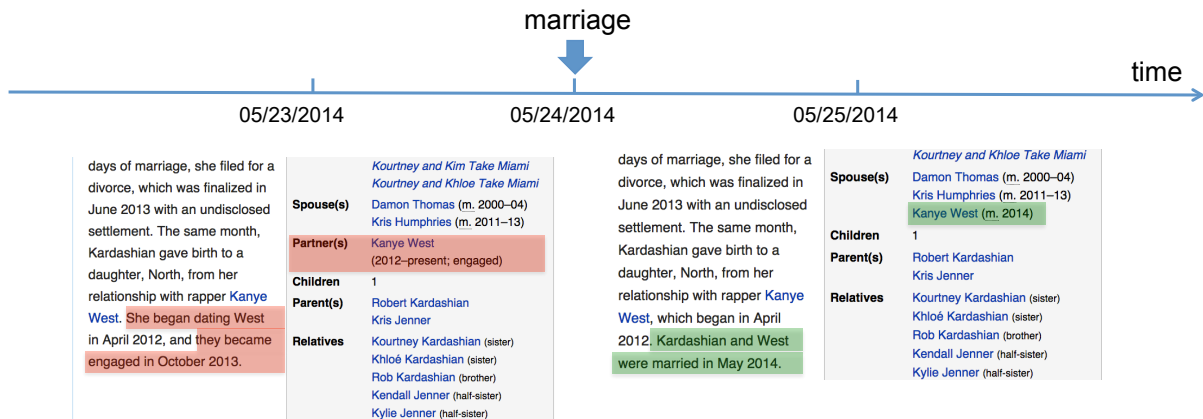


Figure 1: A snapshot of Kim Kardashian’s Wikipedia revision history, highlighting text and infobox changes. In red (and green) are the differences between the page on 05/25/2014 and 05/23/2014: things that are deleted from (and added to) the page.

infobox and text changes.

In summary, our contributions are as follows: (1) we present an algorithm that uses Wikipedia edit histories as distantly labeled data to learn which verbs result in which state changes to entities, and experimentally demonstrate its success, (2) we make available this set of distantly labeled training data on our website<sup>1</sup>, and (3) we also make available our learned mappings from verbs to state changes, as a resource for other researchers, on the same website.

## 2 Method

### 2.1 Data Construction

We construct a dataset from Wikipedia edit histories 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/her Wikipedia revision history  $R_p$  has a set of ordered dates  $T_p$  on which revisions are made to his/her Wikipedia page (we consider date granularity). Each revision  $r_{p,t} \in R_p$  is his/her Wikipedia page at date  $t$  where  $t \in T_p$ .

Each Wikipedia revision  $r_{p,t}$  is a set of infobox slots  $S_{p,t}$  and textual content  $C_{p,t}$ . Each infobox slot  $s \in S_{p,t}$  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 holds in reality.

A document  $d_{p,t}$  in our data set is the *difference*<sup>2</sup> between any two consecutive revisions separated by more than 24 hours i.e.,  $d_{p,t} = r_{p,t+2} - r_{p,t}$ , where  $r_{p,t+2}$  is the *first* revision on date  $t + 2$  and  $r_{p,t}$  is the *last* revision on date  $t$  (as a page can be revised many times in a day).

A document  $d_{p,t}$  is therefore a set of infobox changes  $\Delta S_{p,t}$  and textual changes  $\Delta C_{p,t}$ . Each slot change  $\delta s \in \Delta S_{p,t} = \langle s_{att}, \delta s_{value}, \delta s_{start}, \delta s_{end} \rangle$  is prefixed with + or – to indicate whether they are added or deleted in  $r_{p,t+2}$ . Similarly, each text change  $\delta c \in \Delta C_{p,t}$  is prefixed with + or – to indicate whether they are added or deleted.

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:  $\langle \text{SPOUSE}, +\text{“Kanye West”}, +\text{“2014”}, \text{“ ”} \rangle$ ,  $\langle \text{PARTNER}, -\text{“Kanye West”}, -\text{“2012-present; engaged”}, \text{“ ”} \rangle$  and a set of text changes:  $+\text{“Kardashian and West were married in May 2014”}$ ,  $-\text{“She began dating West”}$ ,  $-\text{“they became engaged in October 2013”}$ .

For each  $d_{p,t}$ , we use  $\Delta S_{p,t}$  to label the document and  $\Delta C_{p,t}$  to extract features for the document. We label  $d_{p,t}$  that has a new value or start time added to its infobox:  $\langle s_{att}, +\delta s_{value}, *, * \rangle \in \Delta S_{p,t}$  or  $\langle s_{att}, *, +\delta s_{start}, * \rangle \in \Delta S_{p,t}$  with the label *begin-satt* and label  $d_{p,t}$  that has a new end time added to its infobox:  $\langle s_{att}, *, *, +\delta s_{end} \rangle \in \Delta S_{p,t}$  with the label *end-satt*.

The label represents the state change that

<sup>1</sup><http://www.cs.cmu.edu/~dwijaya/postcondition.html>

<sup>2</sup>a HTML document obtained by “compare selected revisions” functionality in Wikipedia

happens in  $d_{p,t}$ . For example, in Figure 1,  $d_{kim, 05/23/2014}$  is labeled with *begin-spouse*.

The revision history dataset that we make available for future research consists of all documents  $d_{p,t}$ , labeled and unlabeled,  $\forall t \in T_p$ ,  $t \in [01/01/2007, 12/31/2012]$ , and  $\forall p \in P$ ; a total of 288,184 documents from revision histories of 16,909 Wikipedia entities. Using our labeling process, we find that out of 288,184 documents, only 41,139 have labels (i.e., have their infobox updated with new values/start/end time). The distribution of labels in the dataset is skewed towards birth and death events as these are life events that happen to almost all person entities in Wikipedia. The distribution of labels in the dataset that we release can be seen in Figure 2. We show only labels that we evaluate in our task.

For our task of learning state changing verbs from this revision history dataset, for each labeled  $d_{p,t}$ , we extract as features, verbs (or verbs+prepositions)  $v \in \Delta C_{p,t}$  of which its subject (or object) matches the Wikipedia entity  $p$  and its object (or subject resp.) matches an infobox value, start or end time:  $(v_{subject}, v_{object}) = (arg1, arg2)$  or  $(v_{subject}, v_{object}) = (arg2, arg1)$ , where  $arg1 = p$  and  $\langle s_{att, arg2}, *, * \rangle$  or  $\langle s_{att}, *, arg2, * \rangle$  or  $\langle s_{att}, *, *, arg2 \rangle \in \Delta S_{p,t}$ . We use Stanford CoreNLP (Manning et al., 2014) to dependency parse sentences and extract the subjects and objects of verbs. We find that 27,044 out of the 41,139 labeled documents contain verb edits, but only 4,735 contain verb edits with two arguments, where one argument matches the entity and another matches the value of the infobox change. We use the latter for our task, to improve the chance that the verb edits used as features are related to the infobox change.

## 2.2 Model

We use a Maximum Entropy (MAXENT) classifier<sup>3</sup> given a set of training data =  $\{(\mathbf{v}_{d_\ell}, y)\}$  where  $\mathbf{v}_{d_\ell} = (v_1, v_2, \dots, v_{|V|}) \in R^{|V|}$  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  $\mathbf{w} = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{|Y|}\}$ ,  $\mathbf{w}_y \in R^{|V|}$ , one for each label  $y \in Y$ , the set of all

<sup>3</sup>We use MALLET implementation of MAXENT: <http://mallet.cs.umass.edu/>

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 such as *birthplace*, *birthname*, etc. We manually specified these two types of constraints to all pairs infoboxes where they apply. We have 10 mutex constraints and 23 simultaneously updated constraints. The full list of constraints can be found in our website.

Given a set of constraints, a set of labels  $Y$ , and a set of base verbs<sup>4</sup>  $B$  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  $\{P(y|b) = \text{count}(y, b) / \sum_{y'} \text{count}(y', b)\}$  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  $a_b^y \in \{0, 1\}$  represents whether or not  $b$  should be a feature for  $y$ .

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.

<sup>4</sup>The verb root or base form of a verb (after removing preposition)

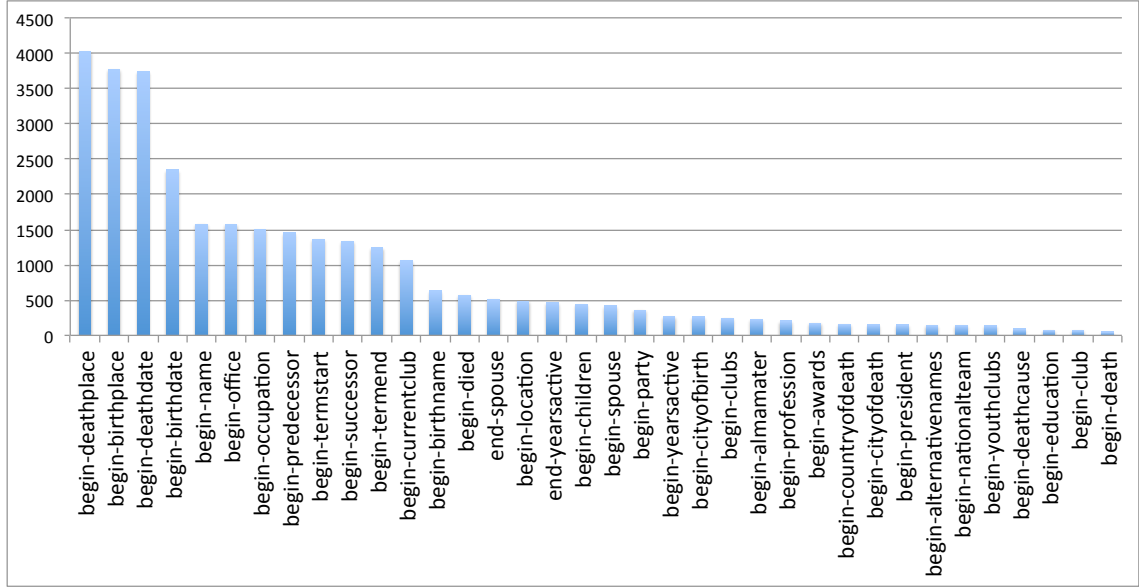


Figure 2: Distribution of labels we evaluate in our task in the revision history dataset.

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  $b = \text{base form of } v_i$ .

After we output  $\mathbf{a}_b$  for each  $b$ , we select features for 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  $b = \text{the base form of } v_i$ . Essentially for each label, we select verb features that have positive weights and are consistent for the label.

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

### 3 Experiments

We use 90% of our labeled documents that have verb edits as features (section 2.1) as training data 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 change) of the document given its verbs features. We compute precision, recall,

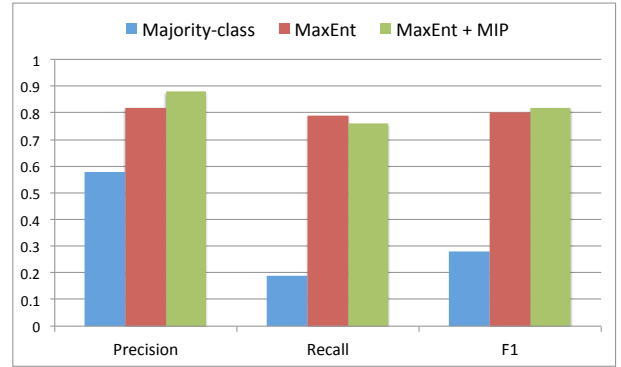


Figure 3: Results of predicting state change labels (infobox types) using verb features.

and F1 values of our predictions and compare the values before and after feature selection (Fig. 3).

To the best of our knowledge, the task to learn state-changing verbs in terms of states defined in existing knowledge bases and learning it from Wikipedia edit histories is novel. There is no previous approach that can be used as baseline; therefore we have compared our structured prediction using MIP and MAXENT with a majority class baseline. Both our approaches (MAXENT and MAXENT + MIP) perform better than the majority class baseline (Figure 3).

We observe the value of doing feature selection by asserting constraints in an MIP formulation. Feature selection improves precision; resulting in a better F1. By asserting constraints,

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

Table 1: Comparison of verb phrases learned before and after feature selection for various labels (infobox types). The texts in bold are (preposition+) noun that occur most frequently with the ⟨verb phrase, label⟩ pair in the training data.

some of the inconsistent verb features for the labels were removed. For example, before feature selection, the verbs: “marry”, 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). In average, we have about 18 verbs per infobox state change in our state changing verb resource that we make available for future research.

## 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 KB 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 to improve KB freshness. This is important because existing KBs are mostly static. We wish to also explore the application of the learned verb resource to domains other than Wikipedia infobox and text e.g., for predicting state changes in the knowledge base from news text.

In this paper, we learned post-conditions of verbs: state changes that occur when an event expressed by a verb happens. As future work we would also explore the feasibility of learning pre-conditions of verbs from Wikipedia revisions. Additionally, most Wikipedia revisions only have text changes without the associated infobox change. Therefore, another line of future work is to also learn from these unlabeled documents.

## Acknowledgments

We thank members of the NELL team at CMU for their helpful comments. This research was supported by DARPA under contract number FA87501320005.

## References

- Aoife Cahill, Nitin Madnani, Joel Tetreault, and Diane Napolitano. 2013. Robust systems for preposition error correction using wikipedia revisions. 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. 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.
- Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.
- 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 wikipedia 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.