

# Language-Aware Truth Assessment of Fact Candidates

**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

This paper introduces FactChecker, language-aware approach to truth-finding. FactChecker differs from prior approaches in that it does not rely on iterative peer voting, instead it leverages language to infer believability of fact candidates. In particular, FactChecker makes use of linguistic features to detect if a given source objectively states facts or is speculative and opinionated. To ensure that fact candidates mentioned in similar sources have similar believability, FactChecker augments objectivity with a co-mention score to compute the overall believability score of a fact candidate. Our experiments on various datasets show that FactChecker yields higher accuracy than existing approaches.

## 1 Introduction

Truth-finding algorithms aim to separate true statements (facts) from false information. More specifically, given a set of statements whose truthfulness is unknown (*fact candidates*), the key goal of truth-finding algorithms is to generate a ranking such that true statements are ranked ahead of false ones. Truth-finders have the potential to address a major obstacle on the Web: the problem of sources spreading inaccurate and conflicting information. This problem continues to grow with the development of tools for easy Web authorship. Blogs, forums and social networking websites are not subject to traditional journalistic standards. Consequently, the accuracy of information reported by these sources is often unclear. Even more established newspapers and websites may sometimes report false information as they race to break stories. Therefore, truth-finding is becoming an in-

creasingly important problem. Information extraction projects aim to distill relational facts from natural language text (Auer et al., 2007; Bollacker et al., 2008; Carlson et al., 2010; Fader et al., 2011; Nakashole et al., 2011; Del Corro and Gemulla, 2013). These projects have produced knowledge bases containing many millions of relational facts between entities. However, despite these impressive advances, there are still major limitations regarding precision. Within the context of information extraction, fact extractors assign confidence scores to extracted facts. However, such scores are often tied to the extractor’s ability to *read* and *understand* natural language text. This is different from a score that indicates the degree to which a given fact candidate is *believable*. Such a believability score is sometimes also referred to as a credibility score or truthfulness score. The believability score reflects the likelihood that a given statement is true. Truth-finding algorithms aim to compute this score for each fact candidate.

Prior truth-finding methods are mostly based on iterative voting, where votes are propagated from sources to fact candidates and then back to sources (Yin et al., 2007; Galland et al., 2010; Pastermack and Roth, 2010; Li et al., 2011; Yin and Tan, 2011). At the core of iterative voting is the assumption that candidates mentioned by many sources are more likely to be true. However, additional aspects of a source influence its trustworthiness, besides external votes.

Our goal is to accurately assess truthfulness of fact candidates by taking into account the language of sources that mention them. A Mechanical Turk study we carried out revealed that there is a significant correlation between objectivity of language and trustworthiness of sources. Objectivity of language refers to the use of neutral, impartial language, which is not personal, judgmental, or emotional. Trustworthiness refers to

a source of information being reliable and truthful. We use linguistics features to detect if a given source objectively states facts or is speculative and opinionated. Additionally, in order to ensure that fact candidates mentioned in similar sources have similar believability scores, our believability computation model incorporates influence of co-mentions. However, we must avoid falsely boosting co-mentioned fact candidates. Our model addresses potential false boosts in two ways: first, by ensuring that co-mention influence is only propagated to related fact candidates; second, by ensuring that the degree of co-mention influence is determined by the trustworthiness of the sources in which co-mentions occur.

The contribution of this paper is a language-aware truth-finding approach. More precisely, we make the following contributions: (1) *Alternative Fact Candidates*: Truth-finders rank a given fact candidate with respect to its alternatives. For example, alternative places where Barack Obama could have been born. Virtually all existing truth-finders assume that the alternatives are provided. In contrast, we developed a method for generating alternative fact candidates. (2) *Objectivity-Trustworthiness Correlation*: We hypothesize that objectivity of language and trustworthiness of sources are positively correlated. To test this hypothesis, we designed a Mechanical Turk study. The study showed that this correlation does in fact hold. (3) *Objectivity Classifier*: Using labeled data from the Mechanical Turk study, we developed and trained an objectivity classifier which performed better than prior proposed lexicons from literature. (4) *Believability Computation*: We developed FactChecker, a truth-finding method that linearly combines objectivity and co-mention influence. Our experiments showed that FactChecker outperforms prior methods.

## 2 Fact Candidates

In this section, we formally define what constitutes a fact candidate and describe how we go about understanding semantics of fact candidates. We then present our approach for generating alternative fact candidates.

### 2.1 Representation

The *triple* format is the most common representation of facts in knowledge bases. A formal specifi-

cation of the triple format is presented in the RDF primer<sup>1</sup>. In RDF, data is represented as subject-predicate-object (SPO) triples. In this work, we restrict predicates to *verbs* (or *verbal phrases* such as “plays for”, “graduated from”, etc.). Literature on automatic relation discovery (Fader et al., 2011) has shown that verbal phrases uncover a large fraction of binary predicates while reducing the amount of noisy phrases that do not denote any relations. Therefore, we define a fact candidate as follows:

**Definition 1 (Fact Candidate)** *A fact candidate  $f_i$  is an  $\langle S \rangle V \langle O \rangle$  triple; where  $S$  is the subject,  $V$  is a verbal phrase, and  $O$  is the object. We aim to compute the truthfulness of  $f_i$ ,  $\tau(f_i) \in \{T, F\}$ , where  $T$  and  $F$  stand for true and false, respectively.*

Note that in this paper we are interested in cases where  $\tau(f_i)$  is either  $T$  or  $F$ . That is, we assess truthfulness of *factual* statements and not opinions whose truthfulness is often both  $T$  and  $F$  to some degree. For example, the triples:  $\langle Obama \rangle$  born in  $\langle Kenya \rangle$  and  $\langle Obama \rangle$  graduated from  $\langle Harvard \rangle$  are valid fact candidates. However, the triple:  $\langle Obama \rangle$  deserves  $\langle Nobel Peace Prize \rangle$  is not.

### 2.2 Semantics

Based on the SVO triple, the meaning of a fact candidate can be unclear and ambiguous. Therefore, we first determine the semantics of a fact candidate before computing its truthfulness.

**Entity Types.** We first determine the expected types of the subject and object in the SVO. For example, for the SVO  $\langle Einstein \rangle$  died in  $\langle Princeton \rangle$ , the expected types are *person*  $\times$  *location*. We determine this by first computing the types of entities that are valid for each verb (verbal phrase) in a large SVO collection of 114m SVO triples (Talukdar et al., 2012). Typing verbal phrases is a once-off computation. Our phrase typing method is similar to prior work on typing relational phrases (Nakashole et al., 2012). Examples of typed phrases are:  $\langle person \rangle$  died in  $\langle year \rangle$ ,  $\langle person \rangle$  died in  $\langle location \rangle$ , and  $\langle athlete \rangle$  plays for  $\langle team \rangle$ . Given a triple, we look up the types for the subject and the object and then determine which of the typed phrases are compatible with the current triple. We look up entity types in a knowledge

<sup>1</sup><http://www.w3.org/TR/rdf-primer/>

base containing entities and their types. In particular, we use the NELL entity typing API (Carlson et al., 2010). NELL’s entity typing method has high recall because when entities are not in the knowledge base, it performs on-the-fly type inference using the Web. This is not the case for other options such as (Auer et al., 2007; Bollacker et al., 2008; Hoffart et al., 2011).

**Relation Cardinality.** Next, we learn cardinalities of verbal phrases. Cardinality refers to how arguments of a given relation relate to one another numerically. We define the relation cardinality of a verb  $Card(V)$ , as the average number of expected arguments per given subject. For example, for the relation “died in”, 1 location is expected for each subject. For other relations, the expected number of arguments can be greater than 1 but less than  $n : n \in \mathbb{R}, n > 1$ . We approximate  $n$  using statistics from the 114m SVO corpus based on the average number of arguments per given first argument. In a once-off computation, we generate cardinality approximations per typed verbal phrase  $V$  and its inverse  $V^{-1}$ . For example, we generate the cardinality estimates for both:  $\langle person \rangle$  died in  $\langle location \rangle$  and for  $\langle location \rangle$  INVERSE-OF(died in)  $\langle person \rangle$ .

**Synonymous Relations.** Natural language is diverse. Semantically similar phrases can be syntactically different. Therefore, we learn other verbs that can be used to substitute  $V$  in SVO. We pre-compute synonymous phrases from the 114m SVO corpus using distributional semantics in the same spirit as (Lin and Pantel, 2001; Nakashole et al., 2012).

Synonymous verbs, relation cardinalities, and entity types enable us to generate alternative fact candidates.

### 2.3 Alternative Fact Candidates

Truth-finding methods rank  $f_i$  relative to alternative candidates. While prior methods assume the alternatives are known apriori, we developed a method for generating alternative fact candidates. For a given  $f_i$ , we first identify the *fixed argument*. The fixed argument is the argument of the SVO which when fixed, requires finding the fewest number of alternative candidates. For example, for  $\langle Einstein \rangle$  died in  $\langle Princeton \rangle$ , the solution is to fix the subject. This is because the cardinality of  $\langle person \rangle$  died in  $\langle location \rangle$  is one (1).

On the other hand, the cardinality of “INVERSE-OF(died in)” is many( $n$ ). In other words, the number of places where a person can be born (one) is much fewer than the number of people that can die in a place (many). In our example, alternatives are possible places, other than Princeton, where Einstein could have died. For example:  $\langle Einstein \rangle$  died in  $\langle Germany \rangle$  or  $\langle Einstein \rangle$  died in  $\langle Switzerland \rangle$ . More generally, the fixed argument of fact candidate  $f_i$ , is defined as follows:

**Definition 2 (Fixed Argument)** Let  $Card(V)$  be the cardinality of  $V$  and  $Card(V^{-1})$  be the cardinality of the inverse of  $V$ , if  $Card(V) < Card(V^{-1})$ , then the fixed argument is the subject,  $Arg^{fixed}(f_i) = S$ , else it is the object,  $O$ . If  $Card(V) == Card(V^{-1})$ , then both arguments are fixed, one at a time.

We use the fixed argument to define a *topic* as the fixed argument plus the verb. Therefore, for the SVO  $\langle X \rangle$  died in  $\langle Y \rangle$ , the topic “places where X died”, ( $Arg^{fixed} = S$ ), is not the same as the topic “people who died in Y” ( $Arg^{fixed} = O$ ).

To locate alternatives, we use the topic ( $Arg^{fixed} + V$ ) as a query. We search three sources to either locate relevant documents or relevant triples: the Google Web search API, the 114m SVO collection, and the NELL KB. The SVO collection and the KB return triples, however, the Web search API returns documents. Therefore, we apply a triple extractor to the retrieved documents. For all potential alternative triples, we perform type checking to ensure that the arguments of the triples are type-compatible with  $f_i$ . Furthermore, we generate an additional query for every synonymous verb  $sV_i$ , replacing  $V$  with  $sV_i$ . Example queries are: “Einstein died in”, “Einstein passed in”, etc.

### 3 Objectivity and Trustworthiness

The principle of *objective journalism*, which is a significant part of journalistic ethics, aims to promote factual and fair reporting, undistorted by emotion or personal bias (Schudson, 1978; Kaplan, 2002). Objectivity is also required in reference sources such as encyclopedias, scientific publications, and textbooks. For example, Wikipedia enforces a neutral point-of-view policy (NPOV)<sup>2</sup>. Articles violating the NPOV policy are marked

<sup>2</sup>[http://en.wikipedia.org/wiki/Wikipedia:Neutral\\_point\\_of\\_view](http://en.wikipedia.org/wiki/Wikipedia:Neutral_point_of_view)

to indicate potential bias. While opinions, emotions, and speculations can also be expressed using objective language, they are often stated using subjective language (Turney et al., 2002; Riloff and Wiebe, 2003; Yu and Hatzivassiloglou, 2003; Wiebe et al., 2004; Liu et al., 2005; Recasens et al., 2013). For example, consider the following pieces of text:

*(S) Well, I think Obama was born in Kenya because his grandma who lives in Kenya said he was born there.*

*(O) Theories allege that Obama’s published birth certificate is a forgery, that his actual birthplace is not Hawaii but Kenya.*

Text *S* is a snippet from Yahoo Answers and text *O* is a snippet from the Wikipedia page titled: “Barack Obama Citizenship Conspiracy Theories”. *S* is subjective, expressing the opinion of the author. On the other hand, *O* is objective, stating only what has been alleged. Literature on sentiment analysis (Turney et al., 2002; Liu et al., 2005), subjectivity detection (Riloff and Wiebe, 2003; Wiebe et al., 2004), and bias detection (Yu and Hatzivassiloglou, 2003; Recasens et al., 2013) has developed lexicons for identifying subjective language. Due to the principle of objective journalism and the requirement of objectivity placed on reference sources, we hypothesize a link between objectivity and trustworthiness as follows.

**Hypothesis 1** *Objective sources are more trustworthy than subjective sources. Therefore, we can assume that fact candidates stated in objective sources are more likely to be true than those stated in subjective sources.*

To test the validity of the hypothesis, we carried out a study where we solicited human input.

### 3.1 Mechanical Turk Study

We deployed an annotation study on Amazon Mechanical Turk (MTurk)<sup>3</sup>, a crowd-sourcing platform for tasks requiring human input. Tasks on MTurk are small questionnaires consisting of a description and a set of questions. Our study consisted of two independent tasks. The first task was titled “Trustworthiness of News Articles”, where annotators were given a link to a news article and

<sup>3</sup><http://www.mturk.com>

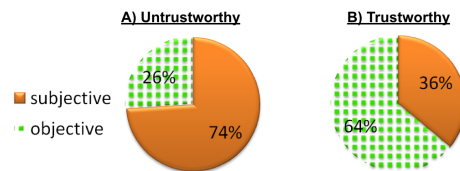


Figure 1: Summary of the results of the annotation study on objectivity and trustworthiness.

asked to judge if they thought it was trustworthy or not. The second task was titled “Objectivity of News Articles”. For this task, annotators were asked to judge if a given article is objective or subjective. For both tasks a third option of “not sure” was provided. We randomly selected 500 news articles from a corpus of about 300,000 news articles obtained from Google News from the topics of Top News, Business, Entertainment, and SciTech. For each task, every article was judged by three annotators. This produced a total of 3000 annotations. When we analyzed the output, we accepted a label as valid for a given article if the label was selected by the majority of the judges. Based on this criteria, we obtained a set of 420 articles that were both labeled for trustworthiness and objectivity.

A summary of the outcome of the study is shown in Figure 1; 74% of the untrustworthy articles were independently labeled as subjective. On the other hand, 64% of trustworthy articles were independently labeled as objective. These results indicate a non-trivial positive correlation between objectivity and trustworthiness. We leverage this correlation in our believability computation model. To incorporate objectivity in FactChecker, we require for a given source document, an objectivity score  $\in [0, 1]$ , where 0 means the source is subjective and 1 means it is objective. Next, describe our method for automatically determining objectivity of sources.

### 3.2 Automatic Objectivity Detection

We trained a logistic regression classifier to predict the objectivity of a document. For training and testing data, we used the labeled data from the Mechanical Turk study. We additionally used labeled text from prior work on subjectivity detection (Pang and Lee, 2004). This resulted in a total of 4,600 documents, half subjective and the other half objective. We used 4000 documents for

#	Feature
1	Subjectivity lexicon of strong and weak subjective words (Riloff and Wiebe, 2003).
2	Sentiment lexicon of positive and negative words (Liu et al., 2005).
3	Wikipedia-derived bias lexicon (Recasens et al., 2013).
4	Part-of-speech (POS) tags
5	Frequent bi-grams

Table 1: Features used for the objectivity detector.

training, 2000 per label. The rest of the documents were split into a development set (380) and a test set (220).

A summary of the features we used is shown in Table 1. Features 1-3 refer to lexicons developed by prior methods on subjectivity (Wiebe et al., 2004), sentiment analysis (Liu et al., 2005) and bias detection (Recasens et al., 2013). Feature 4 refers to part-of-speech tags of the terms found in the document that are also in the lexicons. Feature 5 refers to bi-grams that frequently occur (mention frequency of  $> 10$ ) in the 4,600 documents. The most contributing features were the lexicons, features (1-3) and the frequent bi-grams, feature 5. We discovered that using *frequent* bi-gram features instead of uni-grams or bi-grams resulted in higher precision. The classifier was able to determine that for example bi-grams such as “think that”, “so funny” and “you thought” are negative features for objectivity. Evaluation results of our objectivity detector vs. baselines are shown in Table 2. FactChecker’s objectivity detector has precision of  $0.7814 \pm 0.0539$ , with a 0.9-confidence Wilson score interval (Brown et al., 2001) and this outperforms the baselines. Next, we describe how we leverage objectivity into FactChecker’s truthfulness model.

#### 4 Believability Computation Model

FactChecker computes the believability score of a fact candidate from its: i) objectivity score and (ii) co-mention score. In this section we define each of these scores.

The objectivity score reflects the trustworthiness of sources where a fact candidate is mentioned. Given a fact candidate  $f_i$ , mentioned in a set of documents  $D_i$ , where each document  $d \in$

Approach	Accuracy
Sentiment Lexicon	$0.65 \pm 0.06$
Wikipedia bias Lexicon	$0.69 \pm 0.06$
Subjectivity Lexicon	$0.70 \pm 0.06$
FC-Objectivity Detector	<b><math>0.78 \pm 0.05</math></b>

Table 2: Accuracy of the objectivity detector.

$D_i$  has objectivity  $\mathcal{O}(d)$ ,  $f_i$ ’s objectivity score is defined as follows:

#### Definition 3 (Objectivity Score)

$$\mathcal{O}(f_i) = \log|D_i| \cdot \frac{\sum_{d_k \in D_i} \mathcal{O}(d_k)}{|D_i|} \quad (1)$$

We do not use the sum of objectivity of sources as the objectivity score because this enables fact candidates mentioned in many low objectivity sources to have high aggregate objectivity. Similarly, we avoid using average objectivity of the sources as it overestimates objectivity of candidates stated in few sources. A candidate mentioned in 10 sources with 0.9 objectivity should have higher objectivity than a candidate stated in 1 source of 0.9 objectivity. In Equation 1,  $\log|D_i|$  addresses this issue.

The co-mention score aims to ensure that fact candidates mentioned in similar sources have similar believability scores. Suppose candidate  $f_i$  is mentioned in many highly objective sources, another candidate  $f_j$  is stated in only one highly objective source  $d_k$  where  $f_i$  is also mentioned. Then the believability of  $f_j$  should be boosted by it being co-mentioned with  $f_i$ . If on the other hand  $f_i$  and  $f_j$  were co-mentioned in a subjective source,  $f_j$  should receive less boost from  $f_i$ . This leads us to the co-mention score  $\mu(f_i)$  of a candidate.

#### Definition 4 (Co-Mention Score)

$$\mu(f_i) = \rho(f_i) + \sum_{f_j \in \mathcal{F}} w_{ij} \mu(f_j) \quad (2)$$

Where  $\rho(f_i)$  is the normalized mention frequency of  $f_i$ . The propagation weight  $w_{ij}$  controls how much boost is obtained from a co-mentioned candidate. We define propagation weight,  $w_{ij}$ , as the average of the objectivity of the sources that mention both candidates.

$$w_{ij} = \text{average } \mathcal{O}(d_k) : d_k \in (D_i \cap D_j) \quad (3)$$

where  $\mathcal{O}(d_k)$  is the objectivity of document  $d_k$ ,  $D_i$  and  $D_j$  are the sets of documents that mention  $f_i$  and  $f_j$ , respectively. Notice that we could boost co-mentioned but not related candidates, thereby causing false boosts. To remedy this, we only allow  $w_{ij}$  to be greater than *zero* if the fact candidates  $f_i$  and  $f_j$  are on the same topic. Recall that the topic is determined by the fixed argument (Definition 2) and the verb. Allowing only fact candidates on the same topic to influence each other is important considering that many trivial facts are often repeated in sources of diverse quality.

To leverage the inter-dependencies among *related* co-mentioned fact candidates, we model the solution with a graph ranking method. Each fact candidate is a node and there is an edge between each pair of related fact candidate nodes  $f_i$  and  $f_j$ , with  $w_{ij}$  as the edge weight. Thus, equation 2 can be reformulated as  $\mu = M\mu$ , where  $\mu$  is the co-mention score vector and  $M$  is a Markov matrix which is stochastic, irreducible and aperiodic. Thus, a power method will converge to a solution in a similar manner to PageRank. Implementation consists of iteratively applying Equation 2 until the change in the score is less than a threshold  $\epsilon$ . The solution is the final co-mention scores of fact candidates.

Finally, to compute the believability score of a fact candidate, we linearly combine its objectivity score with its co-mention as follows:

**Definition 5 (Believability Score)**

$$\beta(f_i) = \lambda\mathcal{O}(f_i) + (1 - \lambda)\mu(f_i) \quad (4)$$

Where  $\lambda$  is a weighting parameter  $\in [0, 1]$  which controls the relative importance of the two aspects of FactChecker. As we show in our experiments,  $\lambda$  can be robustly chosen within the range of 0.2 to 0.6. In our experiments we used  $\lambda = 0.6$ .

The entire procedure of FactChecker is summarized in Algorithm 1.

**5 Evaluation**

We evaluated FactChecker for accuracy. We define accuracy as the probability of a true fact candidate having a higher believability score than a false candidate. Let  $\tau(f_i) \in \{T, F\}$  be the truthfulness of a fact candidate  $f_i$ , accuracy is defined as:

---

**Algorithm 1** FactChecker

---

**Input:** A set  $\mathcal{F}$  of fact candidates  
**Input:** KB  $\mathcal{K}$ , SVO corpus  $\mathcal{C}$ , Web  $\mathcal{W}$   
**Output:** A set  $\mathcal{L}$  of rankings  $\forall f_i \in F$   
 $\mathcal{L} = \emptyset$   
**while**  $\mathcal{F} \neq \emptyset$  **do**  
    pick  $f_i$  from  $\mathcal{F}$   
     $\mathcal{A} = \text{getAlternatives}(f_i, \mathcal{K}, \mathcal{C}, \mathcal{W})$   
    PriorityQueue  $L_i = \emptyset$   
    **for all** alternative fact candidates  $f'_j \in \mathcal{A}$  **do**  
         $\beta(f'_j) = \text{getBelievabilityScore}(f'_j)$   
         $L_i.\text{insert}(f'_j, \beta(f'_j))$   
    **end for**  
     $\beta(f_i) = \text{getBelievabilityScore}(f_i)$   
     $L_i.\text{insert}(f_i, \beta(f_i))$   
     $\mathcal{L} \cup L_i$   
    Remove  $f_i$  from  $F$   
**end while**  
**return**  $\mathcal{L}$

---

$$Acc = \frac{\sum_{(\tau(f_i)=T:\tau(f_j)=F)} (\beta(f_i) > \beta(f_j))}{|\{\forall(f_i, f_j) : \tau(f_i) = T \wedge \tau(f_j) = F\}|}$$

**Datasets.** We evaluated FactChecker on three datasets: **i) KB Fact Candidates:** The first dataset consists of fact candidates taken from the fact extraction pipeline of a state-of-the-art knowledge base, NELL (Carlson et al., 2010). The fact candidates span four different relation types: company acquisitions, book authors, movie directors and athlete teams. For each fact candidate, we applied our alternative candidate generation method. We only considered fact candidates with non-trivial alternative candidate sets; where the alternative candidate set is greater than *zero*. Since all of the baselines we compared against assume alternatives are provided, we apply all methods to the same set of alternative fact candidates discovered by our method. Details of this dataset are shown as rows starting with “KB-” in Table 3.

**ii) Wikipedia Fact Candidates:** For the second dataset, we did not restrict the fact candidates to specific topics from a knowledge base, instead we aimed to evaluate all fact candidates about a given entity. We selected entities from Wikipedia. For this, we chose US politicians: all current state senators, all current state governors, and all 44 presidents. First, we extracted fact candidates

	#Candidates	#Alternatives
KB-Acquisitions	50	241
KB-Authors	50	295
KB-Directors	50	228
KB-Teams	40	162
WKP Politicians	54	219
GK Quiz	18	72

Table 3: Fact candidate datasets.

from the infoboxes of the Wikipedia pages of the entities. Second, we applied our alternative candidate generation method to discover alternatives from the Web, SVO corpus, and NELL. Details of the resulting dataset are shown in the row “WKP Politicians” in Table 3.

**iii) General Knowledge Quiz:** The third dataset consists of questions from a general knowledge quiz<sup>4</sup>. We selected questions from the inventions category. Questions are multiple choice, with 4 options per question. Thus, from each question, we created one fact candidate and 3 alternative candidates. Details of the resulting dataset are shown in the row “WKP Quiz” in Table 3.

**Baselines.** We compared FactChecker against five baselines: **i) Vote** counts the number of sources that mention the fact candidate. **ii) TruthFinder** is an iterative voting approach where votes are propagated from sources to fact candidates and then back to sources. Implemented as described in (Yin et al., 2007). **iii) Investment** is also based on transitive voting, however scores are updated differently. A source gets a vote of trust from each candidate it “invests” in, but the vote is weighted by the proportion of trust the source previously “invested” in the candidate relative to other investors. Implemented as described in (Pasternack and Roth, 2010). **iv) PooledInvest** is a variation of investment, we report both because in their paper, there was no clear winner among the two variations. **v) 2-Estimates** is a probabilistic model which approximates error rates of sources and fact candidates (Galland et al., 2010).

### 5.1 Accuracy on KB Fact Candidates

Figure 2 shows accuracy on KB fact candidates. FactChecker achieves accuracy between 70% and 88% and is significantly more accurate than the

<sup>4</sup><http://www.indiabix.com/general-knowledge/>

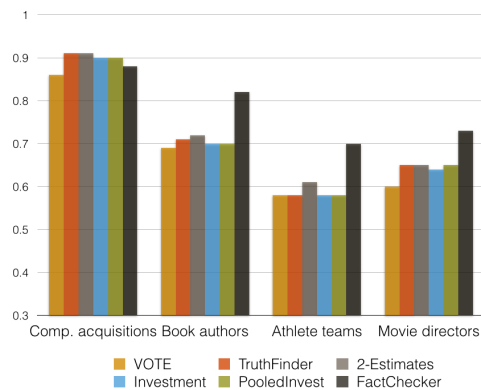


Figure 2: Accuracy of KB fact candidates.

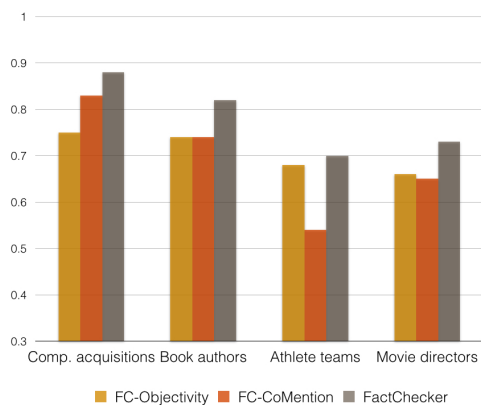


Figure 3: FactChecker variations.

other approaches on all relations except company acquisitions. On book authors, movie directors, and athlete teams, FactChecker outperforms all other approaches by at least 10%, 9%, and 8% respectively. On company acquisitions, the different methods achieve similar accuracy, with TruthFinder being the most accurate and FactChecker is 4% behind. Company acquisitions also yield the lowest difference between Vote and the highest performing method, of 6%. For book authors, movie directors, and athlete teams, the difference between majority Vote and the highest performing method (FactChecker in this case) is 13%, 12%, and 13% respectively.

### 5.2 Accuracy of FactChecker Variations

To quantify how various aspects of our approach affect overall performance, we studied two variations. The first variation is *FC-Objectivity* which only uses objectivity to compute believability. Thus,  $\lambda = 1$  in Definition 5. The second variation is *FC-CoMention* which only uses co-mention scores to compute believability,  $\lambda = 0$ . The

Approach	WKP Politicians	GK Quiz
Vote	0.85±0.09	0.82±0.15
TruthFinder	0.85±0.09	0.82±0.15
2-Estimates	0.85±0.09	0.82±0.15
Investment	0.86±0.08	0.82±0.15
PooledInvest	0.85±0.09	0.82±0.15
FC-Objectivity	0.88±0.08	<b>0.87±0.12</b>
FC-CoMention	0.85±0.09	0.72±0.18
FactChecker	<b>0.90±0.07</b>	<b>0.87±0.12</b>

Table 4: Accuracy on politicians and quiz data sets

last variation is the full FactChecker method using both objectivity and co-mentions with  $\lambda = 0.6$ . From Figure 3, it is clear that both the objectivity of sources and the influence of co-mentions contribute to the overall accuracy of FactChecker. Full-fledged FactChecker performs better than both variations. In most cases, *FC-Objectivity* performs better than *FC-CoMention*.

### 5.3 Accuracy on Wikipedia Fact Candidates

Table 4, column “WKP Politicians”, shows accuracy on Wikipedia fact candidates, with a 0.9-confidence Wilson score interval (Brown et al., 2001). For this dataset we again see FactChecker outperforming the other methods under comparison. On this dataset, FactChecker has a accuracy of  $0.9 \pm 0.07$  and a 5% accuracy advantage over the other methods. The second best performance comes from the *FC-Objectivity* variation, with accuracy of  $0.88 \pm 0.08$ .

### 5.4 Accuracy on General Knowledge Quiz

Table 4, column “GK Quiz”, shows accuracy on the general knowledge quiz fact candidates. On this dataset, FactChecker and its objectivity-only variation (FC-objectivity) have the highest accuracy of 87%. Notice that this dataset was the only one where we did not generate the alternative fact candidates. Instead, we took the options of the multiple choice questions as alternatives. Since the quiz is meant to be taken by humans, the alternatives are often very close, plausible answers. Yet even in this difficult setting, we see FactChecker outperforming the baselines.

Sample fact candidates, with ranked alternatives from all three datasets are shown in Table 5.

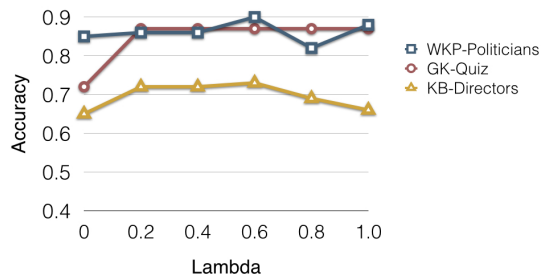


Figure 4: Effect of  $\lambda$  of FactChecker.

## 5.5 Parameter Sensitivity

We analyzed the effect of the selection of lambda  $\lambda$  (see Definition 5) on FactChecker’s performance. The result of this analysis is shown in Figure 4. FactChecker is insensitive to this parameter when  $\lambda$  is varied from 0.2 to 0.6. Therefore, lambda can be robustly chosen within this range.

## 5.6 Discussion

Overall, from these results we make the following observations: i) Majority vote is a competitive baseline; ii) Iterative voting-based methods provide slight improvements on majority vote. This is due to the fact that at the core of iterative voting is still the assumption that fact candidates mentioned in many sources are more likely to be true. Therefore, for both majority vote and iterative voting, when mention frequencies of various alternatives are the same, accuracy suffers. Based on these observations, it is clear that truth-finding solutions need to incorporate fine-grained content-aware features outside of external votes. FactChecker takes a step in this direction by incorporating the document-level feature of objectivity.

## 6 Related Work

There is a fairly small body of work on truth-finding (Yin et al., 2007; Galland et al., 2010; Pasternack and Roth, 2010; Li et al., 2011; Yin and Tan, 2011; Zhao et al., 2012; Pasternack and Roth, 2013). The method underlying most truth-finding algorithms is iterative transitive voting (Yin et al., 2007; Galland et al., 2010; Pasternack and Roth, 2010; Li et al., 2011). Fact candidates are initialized with a score. Trustworthiness of sources is then computed from the believability of the fact candidates they mention. In return, believability of candidates is recomputed based on the trustworthi-



Dataset	Fact Candidate	Alternatives & Ranking
WKP	<i>George W. Bush</i> lived in <i>Midland, TX</i>	1. Midland, TX 2. Compton, CA 3. Washington D.C. 4. Venezuela*
KB	<i>Dirk Kuyt</i> plays for <i>Liverpool</i>	1. Liverpool 2. Cardiff City* 3. Netherlands 4. Hungary*
Quiz	<i>Bifocals</i> invented by <i>Benjamin Franklin</i>	1. Benjamin Franklin 2. Rudolf Diesel* 3. Thomas Alva Edison* 4. Alfred B. Nobel*

Table 5: Sample rankings by FactChecker, alternatives marked (\*) are false. The ranking of the candidate from the “KB” dataset is not completely accurate.

ness of their sources. This process is repeated over several iterations until convergence. (Yin et al., 2007) was the first to implement this idea, subsequent work improved upon iterative voting in several directions. (Dong et al., 2009) incorporates copying-detection; giving high trust to sources that are independently authored. (Galland et al., 2010) approximates error rates of sources and fact candidates. (Pasternack and Roth, 2010) introduces prior knowledge in the form of linear programming constraints in order to ensure that the truth discovered is consistent with what is already known. (Yin and Tan, 2011) introduces supervision by using ground truth facts so that sources that disagree with the ground truth are penalized. (Li et al., 2011) uses search engine APIs to gather additional evidence for believability of fact candidates. WikiTrust (Adler and Alfaro, 2007) is a content-aware but domain-specific method. It computes trustworthiness of wiki authors based on the revision history of the articles they have authored. Motivated by interpretability of probabilistic scores, two recent papers addressed the truth-finding problem as a probabilistic inference problem over the sources and the fact candidates (Zhao et al., 2012; Pasternack and Roth, 2013). Truth-finders based on textual entailment such as TruthTeller (Lotan et al., 2013) determine if a sentence states something or not. The focus is on understanding natural language, including the use of negation. This is similar to the goal of fact extraction (Banko et al., 2007; Carlson et al., 2010; Fader et al., 2011; Nakashole et al., 2011; Del Corro and Gemulla, 2013).

In a departure from prior work, our method leverages language of sources in its believability

computation model. Furthermore, we introduced a co-mention score which is designed to avoid potential false boots among fact candidates. Additionally, we developed a method for generating alternative fact candidates. Prior methods assume these are readily available. Only (Li et al., 2011) uses the Web to identify alternatives, however, this is only done after manually specifying the fixed argument. In contrast, we introduced a method for identifying the fixed argument based on relation cardinalities learned from SVO statistics.

## 7 Conclusion

In this paper, we presented FactChecker, a language-aware approach to truth-finding. In contrast to prior approaches, which rely on external votes, FactChecker includes objectivity of sources in its believability computation model.

FactChecker can be seen as a first step towards language-aware truth-finding. Future directions include using more sentence-level features such the use of hedges, assertive verbs, and factive verbs. These types of words fall into a class of words used to express certainties, speculations or doubts — these are important cues that FactChecker can leverage.

## 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

- S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z.G. Ives: DBpedia: A Nucleus for a Web of Open Data. In *Proceedings of the 6th International Semantic Web Conference (ISWC)*, pages 722–735, Busan, Korea, 2007.
- B. T. Adler, L. de Alfaro: A content-driven reputation system for the wikipedia. In *Proceedings of the 16th International Conference on World Wide Web (WWW)*, pages 261–270, 2007.
- M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, O. Etzioni: Open Information Extraction from the Web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2670–2676, Hyderabad, India, 2007.
- K. D. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor: Freebase: a Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages, 1247–1250, Vancouver, BC, Canada, 2008.
- L. D. Brown, T.T. Cai, A. Dasgupta: Interval Estimation for a Binomial Proportion. *Statistical Science* 16: pages 101–133, 2001.
- E. Cabrio, S. Villata: Combining Textual Entailment and Argumentation Theory for Supporting Online Debates Interaction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 208–212, 2012.
- A. Carlson, J. Betteridge, R.C. Wang, E.R. Hruschka, T.M. Mitchell: Coupled Semi-supervised Learning for Information Extraction. In *Proceedings of the Third International Conference on Web Search and Web Data Mining (WSDM)*, pages 101–110, New York, NY, USA, 2010.
- L. Del Corro, R. Gemulla: ClausIE: clause-based open information extraction. In *Proceedings of the 22nd International Conference on World Wide Web (WWW)*, pages 355–366. 2013.
- X. Dong, L. Berti-Equille, D. Srivastava: Truth discovery and copying detection in a dynamic world. In *Proceedings of the VLDB Endowment PVLDB*, 2(1), pp. 562–573, 2009.
- A. Fader, S. Soderland, O. Etzioni: Identifying Relations for Open Information Extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1535–1545, Edinburgh, UK, 2011.
- A. Galland, S. Abiteboul, A. Marian, P. Senellart: Corroborating information from disagreeing views. In *Proceedings of the 3rd International Conference on Web Search and Web Data Mining (WSDM)*, pages 131–140, 2010.
- C. Havasi, R. Speer, J. Alonso: ConceptNet 3: a Flexible, Multilingual Semantic Network for Common Sense Knowledge. In *Proceedings of the Recent Advances in Natural Language Processing (RANLP)*, Borovets, Bulgaria, 2007.
- J. Hoffart, F. Suchanek, K. Berberich, E. Lewis-Kelham, G. de Melo, G. Weikum: YAGO2: Exploring and Querying World Knowledge in Time, Space, Context, and Many Languages. In *Proceedings of the 20th International Conference on World Wide Web (WWW)*, pages 229–232, Hyderabad, India. 2011.
- R. Kaplan: *Politics and the American Press: The Rise of Objectivity*, pages 1865–1920, New York, Cambridge University Press, 2002.
- X. Li and W. Meng, C. T. Yu: T-verifier: Verifying truthfulness of fact statements. In *Proceedings of the International Conference on Data Engineering (ICDE)*, pp. 63–74, 2011.
- D. Lin, P. Pantel: DIRT: discovery of inference rules from text. *KDD* 2001
- B. Liu, M. Hu, J. Cheng: Opinion Observer: analyzing and comparing opinions on the Web. In *Proceedings of the 14th International Conference on World Wide Web (WWW)*, pages 342351, 2005.
- A. Lotan, A. Stern, I. Dagan TruthTeller: Annotating Predicate Truth. In *Proceedings of Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, pp. 752–757, 2013.
- N. Nakashole, M. Theobald, G. Weikum: Scalable Knowledge Harvesting with High Precision and High Recall. In *Proceedings of the 4th International Conference on Web Search and Web Data Mining (WSDM)*, pages 227–326, Hong Kong, China, 2011.
- N. Nakashole, T. Tyenda, G. Weikum: Fine-grained Semantic Typing of Emerging Entities. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1488–1497, 2013.
- N. Nakashole, G. Weikum, F. Suchanek: PATTY: A Taxonomy of Relational Patterns with Semantic Types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1135 – 1145, Jeju, South Korea, 2012.
- V. Nastase, M. Strube, B. Boerschinger, C. Zirn, A. Elghafari: WikiNet: A Very Large Scale Multilingual Concept Network. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC)*, Malta, 2010.
- B. Pang, L. Lee: A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based

- on Minimum Cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 271-278, 2004.
- J. Pasternack, D. Roth: Knowing What to Believe. In *Proceedings the International Conference on Computational Linguistics (COLING)*, pp. 877-885, Beijing, China. 2010.
- J. Pasternack, D. Roth: Latent credibility analysis. In *Proceedings of the 22nd International Conference on World Wide Web (WWW)*, pp. 1009-1020, 2013.
- E. Riloff, J. Wiebe: Learning Learning extraction patterns for subjective expressions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1051-1062, 2011.
- M. Recasens, C. Danescu-Niculescu-Mizil, D. Jurafsky: Linguistic Models for Analyzing and Detecting Biased Language. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1650-1659, 2013.
- F. Niu, C. Zhang, C. Re, J. W. Shavlik: DeepDive: Web-scale Knowledge-base Construction using Statistical Learning and Inference. In the VLDS Workshop, pages 25-28, 2012.
- M. Schudson: *Discovering the News: A Social History of American Newspapers*. New York: Basic Books. 1978.
- F. M. Suchanek, M. Sozio, G. Weikum: SOFIE: A Self-organizing Framework for Information Extraction. In *Proceedings of the 18th International Conference on World Wide Web (WWW)*, pages 631-640, Madrid, Spain, 2009.
- P. P. Talukdar, D. T. Wijaya, T.M. Mitchell: Acquiring temporal constraints between relations. In *Proceeding of the 21st ACM International Conference on Information and Knowledge Management*, pages 992-1001, CIKM 2012.
- P. D. Turney: Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 417-424. 2002.
- J. Wiebe, T. Wilson, R. Bruce, M. Bell, M. Martin: Learning subjective language. *Computational Linguistics*, 30(3):277-308. 2004.
- X. Yin, J. Han, P. S. Yu: Truth Discovery with Multiple Conflicting Information Providers on the Web. In *Proceedings of the International Conference on Knowledge Discovery in Databases (KDD)*, pages 1048-1052. 2007.
- X. Yin, W. Tan: Semi-supervised truth discover. In *Proceedings of the 19th International Conference on World Wide Web (WWW)*, pp. 217-226, 2011.
- H. Yu, V. Hatzivassiloglou: Towards Answering Opinion Questions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages. 129-136, 2003.
- B. Zhao, B. I. P. Rubinstein, J. Gemmell, J. Han: A Bayesian approach to discovering truth from conflicting sources for data integration. In *Proceedings of the VLDB Endowment (PVLDB)*, 5(6):550-561, 2012.