Multimodal Learning for Web Information Extraction

Dihong Gong
University of Florida
Gainesville, Florida, United States
gongd@ufl.edu

Daisy Zhe Wang
University of Florida
Gainesville, Florida, United States
daisyw@ufl.edu

Yang Peng
University of Florida Gainesville,
Florida, United States
yangpengsnf@ufl.edu

ABSTRACT

We consider the problem of extracting text instances of predefined categories (e.g. **city** and **person**) from the Web. Instances of a category may be scattered across thousands of independent sources in many different formats with potential noises, which makes open-domain information extraction a challenging problem. Learning syntactic rules like "cities such as _" or "_ is a city" in a semi-supervised manner using a few labeled examples is usually unreliable because 1) high quality syntactic rules are rare and 2) the learning task is usually underconstrained. To address these problems, in this paper we propose to learn multimodal rules to combat the difficulty of syntactic rules. The multimodal rules are learned from information sources of different modalities, which is motivated by an intuition that information that is difficult to disambiguate correctly in one modality may be easily recognized in another. To demonstrate the effectiveness of this method, we have built a sophisticated end-to-end multimodal information extraction system that takes unannotated raw web pages as input, and generates a set of extracted instances (e.g. *Boston* is an instance of **city**) as outputs. More specifically, our system learns reliable relationship between multimodal information by multimodal relation analysis on big unstructured data. Based on the learned relationship, we further train a set of multimodal rules for information extraction. Experimental evaluation shows that a greater accuracy for information extraction can be achieved by multimodal learning.

CCS CONCEPTS

• **Information systems** → *Data extraction and integration*;

KEYWORDS

Multimodal, Information Extraction, Web Mining

ACM Reference Format:
Dihong Gong, Daisy Zhe Wang, and Yang Peng. 2017. Multimodal Learning for Web Information Extraction. In *Proceedings of MM '17, Mountain View, CA, USA, October 23–27, 2017,* 9 pages.
https://doi.org/10.1145/3123266.3123296

MM '17, October 23–27, 2017, Mountain View, CA, USA

ACM ISBN 978-1-4503-4906-2/17/10…$15.00
https://doi.org/10.1145/3123266.3123296

1 INTRODUCTION

The task of information extraction has been traditionally defined as extracting information from unstructured or semi-structured text in the form of text strings which are placed into slots labeled to indicate the kind of information that can fill them. For example, **city** is an example slot (or category), and *Boston* is an example instance that can fill the slot **city**. While wrapper [9] is one of the most popular methods used to extract information from semi-structured text, it fails when the text becomes less structured. The proliferation of the Web, which largely consists of unstructured documents lacking semantic metadata, advocates the development of methods that can efficiently extract information from less structured text.

Supervised machine learning approaches such as [2, 29] have been shown to be effective for extracting information from unstructured text. However, these approaches usually rely on availability of labeled training samples in a large scale. Recent research interest has been focused on semi-supervised learning approaches [1, 3, 21]. The semi-supervised learning algorithms learns iteratively in a bootstrapping manner. They start with a handful of labeled seed instances, then learn syntactic rules based on context of seed instances. The recently learned syntactic rules can be used to extract new instances which are then used to learn new syntactic rules. Though semi-supervised learning methods overcome difficulty of supervised methods and can easily extract knowledge in a scale that was impossible before, they often exhibit unacceptable accuracy. This is primarily caused by: 1) high quality syntactic rules are rare; 2) given limited amount of training data, the learned syntactic rules are usually unable to properly constrain the problem. This motivates us to develop stronger rules to better guide the learning process.

In this paper, we propose a novel learning method that learns multimodal rules to improve the reliability of syntactic rules. The thesis explored in this paper is that, a much higher accuracy for information extraction can be achieved by learning multimodal rules. The major motivation behind our approach is that multimodal information (e.g. text, image, and audio) usually correlates and complements with each other, which allows us to develop more reliable information extractors by making use of information from multiple modalities. More specifically, in this paper we have focused on text and image modalities due to their broad research interest and high availability of information. Given text categories of interest, we learn related visual concepts (multimodal relations) by multimodal relation analysis. For example, visual concepts *plane*, *bus*, and *car* are related to text category **vehicle**. Once multimodal relations are calculated, multimodal rules (including text rules and visual rules) can be developed. In our system, the text rules are syntactic rules (e.g. "cities such as _") as usual, but visual rules are based on visual concepts of an instance. In this way, an instance is evaluated based on information from both text and image modalities. Extensive

experiments confirm that, extracting information using multimodal rules can better guide the learning process, resulting in a much greater extraction accuracy.

We believe the contributions of this paper can be summarized as follows.

- We present a novel multimodal learning approach that effectively makes use of information across multiple modalities. To the best of our knowledge, this is the first time that multimodal learning method is applied to extracting text entities from the Web.
- Based on the multimodal learning algorithm, we developed a sophisticated and scalable end-to-end system for multimodal information extraction. Our system takes unannotated raw web pages and handful of seed instances as inputs, then automatically extracts information in a self-supervised manner, minimizing the human intervention. The system is self-contained that includes components of the entire information extraction pipeline: data crawling, HTML web page parsing, indexing, syntactic rule mining and instance extraction, visual object detector training, visual object detection using deep learning, and a web portal for presenting learned knowledge as well as rules. For best scalability, the system is implemented in a distributed manner for crawling (based on Amazon EC2), storage (based on Hadoop HDFS), and information extraction (based on Apache Spark).
- We present an effective approach to learn multimodal relations between text and image. The multimodal relation learning is one of the most important steps in multimodal learning because it bridges concepts over different modalities, which allows information to flow across the modalities.
- Through experimental evaluation on a real-world information extraction task, we demonstrate the effectiveness of the multimodal learning method, and found that multimodal extraction can greatly improve the accuracy over the corresponding unimodal methods.

2 RELATED WORK

Recently, significant progress has been made for mining knowledge from semi-structured or unstructured text in both commercial and academic domains. For example, Google is building the largest knowledge base (a collection of structured text knowledge) called Knowledge Vault [11]. As of 2014, it contained 1.6 billion facts which had been collected automatically from the Internet. In academic domains, many projects such as YAGO [26], NELL [5], DBpedia [14] and DeepDive [18] have attracted unprecedented research attention in the recent years.

In this paper, we shall focus on mining categorical knowledge (e.g. *Boston* is an instance of **city** category) from large corpus of unstructured text. Instances of a category may be scattered across thousands of independent sources in many different formats with potential noises, which makes open-domain information extraction a challenging problem. While programs like wrapper [9] can extract information from well structured text with high precision, they usually fail when the text becomes less structured. Recent research efforts have been focused on developing machine learning models to automatically extract information from unstructured text.

Supervised approaches train machine learning models with a set of labeled training data, and then perform text classification using the models. For example, conditional random field (CRF) can be used for named entity recognition (NER) where CRF models are trained and then applied to extract sets of named entities such as **Person**, **Location** and **Organization** [16]. However, due to the limitation in availability of labeled training samples, the supervised machine learning approaches cannot be easily scaled up to very large knowledge bases with thousands of categories.

Bootstrapping approaches based on semi-supervised learning start with a small number of labeled seed instances and iteratively grow labeled examples by alternatively learning extraction rules and extracting new labeled examples. The advantage of bootstrapping approaches is that they require only a small number of seed instances for each category, which allows them to easily scale up to large knowledge bases. Bootstrapping approaches have been shown to be effective for information extraction from unstructured text. For example, Brin [3] proposed an effective bootstrapping approach to extract a relation of (author,title) pairs from the World Wide Web; Riloff et. al. [21] extracted semantic lexicon and dictionary of extraction patterns using a multi-level bootstrapping method; Agichtein et. al. [1] demonstrated an information extraction system called *Snowball* to extract relational tuples from newspaper documents. However, the bootstrapping doesn't come without it own disadvantages. Accuracy typically declines as iteration increases because errors in labeling accumulate, a problem that has been called *semantic drift* [8].

To reduce the error accumulation, many algorithms have been studied in the literature. Due to the learning problem is usually underconstrained, algorithms that are designed to add additional constraints to the problem can effectively reduce the errors. Coupling is one of the techniques that are used to further constrain the problem [6, 27]. The coupling uses positive examples of one category as negative examples of another, so that instances that are positive examples of other categories are not extracted. Type checking [6, 19] by specifying types (e.g. proper noun, common noun) of a category or relation arguments between categories that instances have to satisfy is another effective technique used to further constrain the problem. Other approaches reduce the errors by combining predictions from multiple extractors [4, 20], to overcome the difficulty of using single extractor (e.g. syntactic rules). Our work is built on top of these ideas, with distinct focus on learning multimodal information for enhanced information extraction performance.

3 MOTIVATING EXAMPLE

In this section we illustrate the core ideas behind our multimodal learning method through a motivating example. Suppose we want to populate the **bird** category with instances, and we have learned two syntactic patterns "wings of the _" and "_ takes flight" that are used to extract instances of **bird**. Scanning through the text corpus, we found that both *Dreamliner* and *Gull* match the given two syntactic patterns. If we rely only on the syntactic patterns, then both of the two instances should be classified as instances of **bird** category with equal confidence. This immediately results in an outlier instance *Dreamliner*, because it is actually a airplane, not

a bird. This is one of many examples that are difficult to correctly classify when using unimodal learning.

Multimodal learning is proposed to combat this kind of disadvantage. In multimodal learning, we not only consider the syntactic patterns (text modality), but also consider the corresponding visual concepts of the instances (image modality). The visual concepts of *Dreamliner* suggest that it is actually a airplane, instead of a bird. Through this example we have seen how multimodal learning can potentially improve the information extraction accuracy.

4 MULTIMODAL LEARNING

In this section, we present our information extraction system based on text and image multimodal learning. Our system comprises three key stages:

(1) **Multimodal Relation Analysis.** To enable multimodal learning, we first learn the relationship between concepts of the text and image modalities. This stage creates a set of relating visual concepts for each predefined text category, which will be used to develop multimodal classification rules in the next stage.
(2) **Learning Multimodal Rules.** Confidence score of an instance is calculated based on the multimodal rules that it matches. A multimodal rule defines how an instance is matched, and what's the confidence score of that instance if a rule is matched. This stage generates a set of useful multimodal rules for information extraction.
(3) **Multimodal Information Extraction.** In the final stage, we apply the learned multimodal rules to extract information from the real-world data.

4.1 Stage 1: Multimodal Relation Analysis

The task of our information extraction system is to populate a predefined text knowledge base with correct instances. In this paper, we focus on text knowledge base with a flat ontology structure, defined as

$$O_T = \{C_k^{(T)} | k = 1, ..., K\}, \quad (1)$$

where superscript T denotes text categories. In our system, the knowledge base is initialized with a small number of seed instances (or entities, we will use entities and instances interchangeably for the rest of paper) for each category, with $K = 24$ categories in total. In the meanwhile, the ontology of image knowledge base is defined in a similar manner as

$$O_V = \{C_j^{(V)} | j = 1, ..., J\}, \quad (2)$$

where superscript V denotes visual concepts. The image knowledge base is initialized using around 250 labeled images for each category from the ImageNet [10], with $J = 102$ visual concepts. Then visual object detectors are trained based on seed instances of image knowledge base for visual object detection. In this paper, since the focus is on extracting text knowledge (keeping image knowledge base unchanged), visual object detectors are fixed once they are trained.

The multimodal relation analysis learns relationship between concepts of the text and image modalities. Mathematically, an multimodal relation R is a binary relation defined on a pair of multimodal knowledge bases O_T and O_V as

$$R = \{(x, y) | x \in O_T \wedge y \in O_V\}. \quad (3)$$

The $R \subseteq O_T \times O_V$ relates two multimodal knowledge bases on concept level. Intuitively, $(x, y) \in R$ if instances of concept x *co-occurs* with instances of concept y with high frequency, where a pair of instances *co-occurs* if they occur under the same context (e.g. the same meta tag in HTML pages). The calculation of R is through these steps:

- **image tagging** assigns each image in a Web page with proper text descriptions.
- **instance visualization** creates links at instance level from text instance to image.
- **instance aggregation** aggregates instance-level links by pairs of multimodal concepts to extract the multimodal relations.

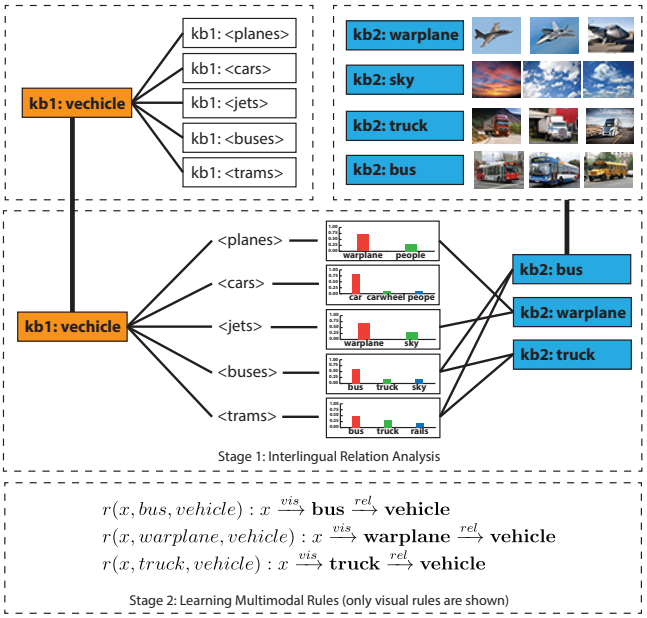

Figure 1: The illustration for multimodal relation analysis (Stage 1) and multimodal rule learning (Stage 2) based on text and image knowledge bases. The kb1 represents for text knowledge base, and kb2 represents image knowledge base. In Stage 1, related visual concepts of text category vehicle are learned by visualizing all instances (planes, cars, etc) of the category, followed by histogram counting, normalization and ranking as described in Section 4.1. In this example, the Stage 1 results in three related visual concepts (bus, warplane,truck) for text category vehicle. In Stage 2, multimodal rules for both text and image modalities are learned as described in Section 4.2. In the figure we illustrate three learned visual rules, where the first rule can be interpreted as "when x has visualization (vis) concept bus, then it can be considered as instance of vehicle because visual concept bus is believed to be related (rel) to vehicle, and the probability of this assertion is given by the rule confidence".

The Figure 1 illustrates the multimodal relation analysis.

4.1.1 Image Tagging. The image tagging program assigns each image a set of noun phrases (tags) that best describe the image [7]. We extract tags of each image based on both image meta and web page context information, following these steps:

- Retrieve top-k most important noun phrases (denoted as NP_k) from a web page containing the target image to be tagged. The importance of a noun phrase t is measured by the following tfidf scoring function

$$tfidf(t,d) = 0.5 + 0.5\frac{f_{t,d}}{\max\{f_{t',d} : t' \in d\}} \times log\frac{N}{n_t}, \quad (4)$$

  where d represents the web page document, $f_{t,d}$ is the frequency of term t in document d, N is the total number of web pages, and n_t is the total number of web pages containing the term t.
- For each noun phrase $t \in NP_k$, if either name of the image (where name is extracted from image url) or *alt* attribute of the <img> tag contains the noun phrases, then assign t as a tag of the image. Note that a single image can have multiple tags. Table 1 shows some example automatically tagged images.

4.1.2 Instance Visualization. At this step, we visualize text instances by linking them to visual concepts. For example, instance *salmon* can be linked to visual concepts such as **fish**. Suppose the image tagging program creates a set of tagged images denoted as

$$\Gamma = \{(I_n, t_1, ..., t_{k_n}) | n = 1, ..., N\}, \quad (5)$$

where I_n denotes the n^{th} image, t_i represents a tag and k_n is the number of tags assigned to the image I_n, and N is the total number of images. Then for each text instance e we retrieve a set of images $I(e)$ whose tags containing the instance, and the visualization score of instance e w.r.t. visual concept y is given by

$$V(e,y) = \frac{1}{P(y)} \sum_{i \in I(e)} \sum_{d \in D(i)} \delta(d,y) P(d), \quad (6)$$

where $D(i)$ denotes a set of detected visual categories on image i (an image can detect multiple objects with different visual concepts), and $P(d)$ is the confidence score that objects of visual concept d exists. The $\delta(d,y)$ is an indicator function that takes value 1 only if $d = y$. The $P(y)$ is the prior probability of detecting an object of visual concept y on any image. The $P(y)$ is used to suppress affect of trivial visual concepts like **people** that have high occurrence frequency. To reject occasional visual linking that is usually unstable, visual category y is considered as visualization of e only if number of images detecting objects in y is above a threshold. We also found that an instance usually have quite limited number of meaningful visualization concepts, so for each instance we retain up to top k visual concepts of the highest visualization scores.

4.1.3 Instance Aggregation. Suppose all instances of text knowledge base used to learn multimodal relation are collectively denoted as E. Then we apply instance visualization program on each instance $e \in E$, and take top $k = 1$ visualization concept for each instance, which results in a set of links

$$L = \{(e,y) | e \in E \wedge y \in O_V\}. \quad (7)$$

The L provides an instance-level linking from text instance to visual concepts. To obtain a concept-level mapping, we aggregate these links by text category, resulting in an aggregation function $A : O_T \times O_V \to \mathbb{R}$.

$$A(x,y) = \frac{1}{|x|} \sum_{e \in x} \delta((e,y) \in L), \quad (8)$$

where $\delta(x)$ is an indicator function takes 1 only if x evaluates True. Then a visual concept y is linked to text category x if $A(x,y)$ is above a threshold. Mathematically, a multimodal relation R is defined as

$$R(x,y) = \{(x,y) | x \in O_T \wedge y \in O_V \wedge A(x,y) \geq \tau\}. \quad (9)$$

The τ represents the minimum percentage of instances in text category x that have visualization concept y.

4.2 Stage 2: Learning Multimodal Rules

The multimodal relation analysis creates relations between concepts across difference modalities. Based on these relations, in this stage we learn multimodal rules for information extraction. In our system, there are two types of multimodal rules: the syntactic rules in text modality and visual rules in image modality.

4.2.1 Learning Syntactic Rules. The syntactic rules are learned from unstructured text corpus in a bootstrapping manner. Suppose we want to learn syntactic rules for category c, given most recently promoted instances E (initially seed instances). Then, syntactic rules of category c are learned in two steps:

(1) **Extract candidate rules.** For each $e \in E$, we extract the preceding words as a candidate rule using the following regular expressions:

[Noun] Verb {JJ|JJR|JJS|IN|DT|CC|TO}+ _ (10)

Noun {JJ|JJR|JJS|IN|DT|CC|TO}+ _ (11)

The _ is a placeholder for the instance e, and the part-of-speech tags are defined by the Penn Treebank Project [15]. The (10) means that a rule can consist of verbs followed by a sequence of adjectives, prepositions, or determiners and optionally preceded by nouns (e.g. "novel written by _"). Alternatively, rule (11) consist of nouns followed by a sequence of adjectives, prepositions, or determiners (e.g. "Google and _"). On the other hand, we also extract words following the instance as candidate rules. More specifically, words following an instance are extracted as a candidate rule if they match these regular expressions:

_ [MD] Verb {DT|IN|CC|TO|JJ|JJR|JJS}* [Noun] (12)

_ [MD] CC Noun (13)

The rule (12) are verbs optionally preceded by modal verb, and optionally followed by a sequence of adjectives, prepositions, or determiners and then nouns (e.g. "_ attended the party"). Once the candidate rules based on E are extracted, they are merged with other candidates extracted in previous iterations to generate a complete candidate list for promotion in the second step.

(2) **Promote top candidate rules.** Syntactic rules are extracted from the top candidate rules of the highest precision. The

precision of a promoted syntactic rule r of category c is estimated by

$$Precision(r, c) = \frac{count(r, c)}{count(r)}, \tag{14}$$

where $count(r, c)$ is the number of distinct matched instances in c, and $count(r)$ is the total number of distinct matched instances. A candidate rule can only be promoted if it matches at least two promoted instances.

4.2.2 Learning Visual Rules. The visual rules recognize instances based on their visualization concepts. A visual rule can be represented as a triple:

$$r(x, V_c, T_c) : x \xrightarrow{vis} V_c \xrightarrow{rel} T_c, \tag{15}$$

which represents if instance x has visualization concept V_c (as described in 4.1.2), and V_c is a related visual concept of text category T_c, then x can be considered as an instance of T_c. For example, instance *Heathrow Airport* has visualization concept **plane** which has been identified as a related visual concept of text category **airport**:

$$Heathrow\ Airport \xrightarrow{vis} \textbf{plane} \xrightarrow{rel} \textbf{airport}$$

Based on this visual rule, the *Heathrow Airport* is considered as instance of **airport**. The precision and recall of a rule $r(x, V_c, T_c)$ is estimated using promoted instances of T_c as positive samples (denoted as s_p), and candidate instances that are candidates or promoted instances of other categories as negative samples (denoted as s_n). Mathematically,

$$Precision(r) = \frac{count(r, s_p)}{count(r)}, \tag{16}$$

and

$$Recall(r) = \frac{count(r, s_p)}{count(s_p)}. \tag{17}$$

The $count(r, s_p)$ represents the number of positive instances matching the rule r, and $count(r)$ is total number of matching instances. The Figure 1 illustrates more example learned visual rules.

4.3 Stage 3: Multimodal Information Extraction

Our system is based on semi-supervised bootstrapping learning that extracts information incrementally. Starting from a handful of seed instances (with each category having 10-20 seeds), we iteratively extract new instances as described in Algorithm 1.

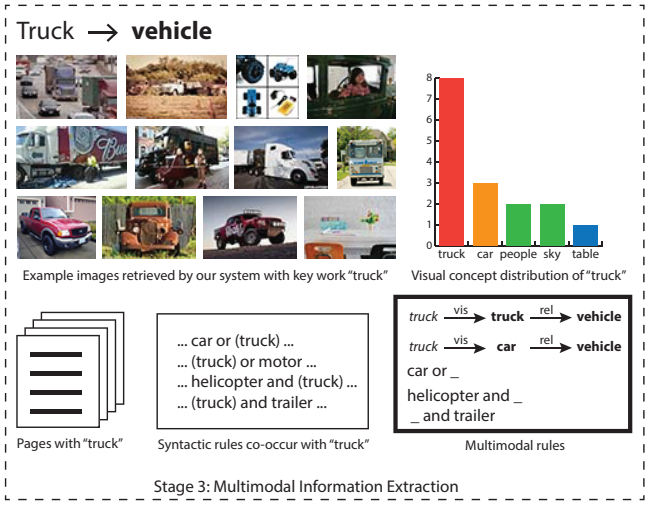

Figure 2: The illustration for extracting "Truck" as an instance of vehicle category. Images whose tags containing "Truck" are retrieved and ranked, and these images are then used to evaluate the related visual concepts (see Section 4.1.2). In the meanwhile, all web pages containing "Truck" are retrieved and then used to calculate the co-occurrence with syntactic rules in the vehicle category. Finally, the confidence score of "Truck" is calculated based on the multimodal rules it matches.

Algorithm 1 Multimodal Information Extraction

Input: The text ontology O_T, image ontology O_V, text corpus and image corpus.
Output: Trusted instances for each text category.
Train visual object detectors using O_V.
for $t = 1, 2, \ldots, \infty$ **do**
  for each category $c \in O_T$ **do**
    Learn multimodal relations;
    Learn syntactic rules R_t;
    Extract candidate instances using R_t;
    Evaluate visual concepts of candidate instances;
    Learn visual rules R_v;
    Evaluate instance confidence with (R_t, R_v);
    Promote top candidate instances;
  end for
end for

The Figure 2 illustrates the multimodal information extraction process. For each category, we maintain two list of candidates: syntactic rule candidates and instance candidates. The maximum size of these two lists are 5,000 and instances of the lowest confidence scores are removed from a list once it overflows. When evaluating visual concepts of an instance, we retain up to top three visual concepts. Finally, the confidence score of an instance is a merge of both syntactic rules and visual rules, given by

$$P(e) = 1 - \prod_{x \in R_t(e)} \prod_{y \in R_v(e)} (1 - P(x))(1 - P(y)), \tag{18}$$

where $R_t(e)$ and $R_v(e)$ are sets of matching syntactic rules and visual rules of e, respectively. The $P(x)$ and $P(y)$ represents the estimated precision of the rules (adjusted to be less than 1.0).

4.4 Coupling and Type Checking

Coupling the learning of syntactic rules by using positive examples of one category as negative examples for others has been shown to be effective in improving the extraction accuracy [6, 21, 27]. We combine this coupling technique into our system. More specifically, at syntactic rule promotion stage, a rule whose precision is less than precision of the same rule in any other categories is not promoted. For instance promotion, we apply a similar coupling technique.

Type checking [6, 19, 22] by specifying types (e.g. proper noun, common noun) of a category or relation arguments between categories that instances have to satisfy is another effective technique used to improve the information extraction accuracy. In our system, we apply type checking on each category by specifying the noun types of instances. For example, instances of **city** can only be proper nouns, and instances of **vehicle** can be either proper nouns or common nouns.

4.5 Large-Scale Implementations

Our multimodal information extraction system is designed to process big data. There are three major components in our system:

- **Mining syntactic rules or instances.** At syntactic rule discovery stage, for given instances we need to retrieve all the matching patterns. This is done by first indexing the location of all noun phrases in the text corpus, then search the given instances and examine the part of speech information of preceding and following words to generate syntactic rules. At the instance discovery stage, for given syntactic rules we retrieve all matching instances. To speedup the retrieval process, we index the text corpus with Trie data structures (that are most suitable for information retrieval as suggested by its name). The core parts of both stages are implemented with C++ for optimal memory and computational efficiency. To scale up the system, we group web pages into blocks (each having around 40,000 web pages), and compute both stages in a distributed manner (based on Apache Spark MapReduce [28]), where data blocks are stored in a distributed file system (based on Hadoop HDFS [24]) providing around 2.0 GB/s peak disk reading performance. Practically, it takes around 3 min for one pass on a text corpus with 100 million web pages on a 52-core cluster.
- **Collecting image corpus.** The image corpus is not included in the Common Crawl data [25] where we derived text corpus. We collect the images for instance visualization (see Section 4.1.2) by extracting the image urls from raw web page data and download the necessary images in a distributed manner using Amazon EC2/S3 [17]. In our system, we download images in a distributed manner with multiple network-independent machines, which provides throughput of around 4 million images per day.
- **Visual object detection.** We train deep learning neural networks based on the state-of-the-art Fast R-CNN algorithm [12]. The neural networks are trained to predict 102 visual concepts, using Cuda GPU (dual nVIDIA Tesla K40C) based on Caffe library [13].

The Figure 3 illustrates the pipeline of our information extraction system.

5 EXPERIMENTS

5.1 Dataset

5.1.1 Ontologies. In our system, we have two ontologies: O_T and O_V, both of which have flat structures. The text ontology O_T consists of 24 different categories, and image ontology O_V has 102 categories. The categories of O_T corresponding to concepts that are both visualizable and not visualizable. The categories of O_V are selected to be potentially related to O_T to efficiently utilize the visual space. Each category of O_T is initialized to have 10-20 seed instances, while each visual concept in O_V are trained with around 250 labeled images of the ImageNet database [10].

5.1.2 Corpus. We derive our text and image corpus based on the Common Crawl dataset [25] that is publicly available on Amazon S3. The entire dataset comprises billions of raw Web pages in warc compressed format, and for our study we take a subset of the data with hundreds of millions of Web pages. These pages are processed following these steps:

(1) Extract the warc files to get raw web pages payload.
(2) Parse the HTML web pages, with a C++ open-source program **gumbo-parser** by Google.
(3) Remove non-English web pages by counting the stop word ratio.
(4) Extract all image urls of each web page, along with *alt* and *src* attributes. We only retain images whose dimension (shortest edge) is at least 150 pixels.
(5) Clean meta and spam from web pages to obtain plain text, then tokenize and apply part-of-speech tagging. The part of speech tagger is based on **Tree-Tagger** [23] for best computational efficiency.
(6) Extract nouns and noun phrases. The nouns can be extracted directly based on part-of-speech information of a word. The noun phrases are extracted based on the following rules:
    - Common noun phrase (e.g. "computer monitor") consist of a sequence of consecutive common nouns;
    - Proper noun phrases (e.g. "National Aeronautics and Space Administration") consist of a sequence of proper nouns connected by optional coordinating conjunction or preposition.

    To remove outliers caused by unreliable part-of-speech tagger, a sequence is considered as as noun or noun phrase only if its frequency is above a threshold.
(7) Assign a set of tags to images by running image tagging program, as described in Section 4.1.1
(8) Crawl web images based on their urls extracted from web pages, and then extract visual objects from these images with trained object detection models. Table 1 shows some example detected objects of **bedroom** category.

This results in corpus of around 100 million web pages, 5 billion tokens, and 150 million images for our study.

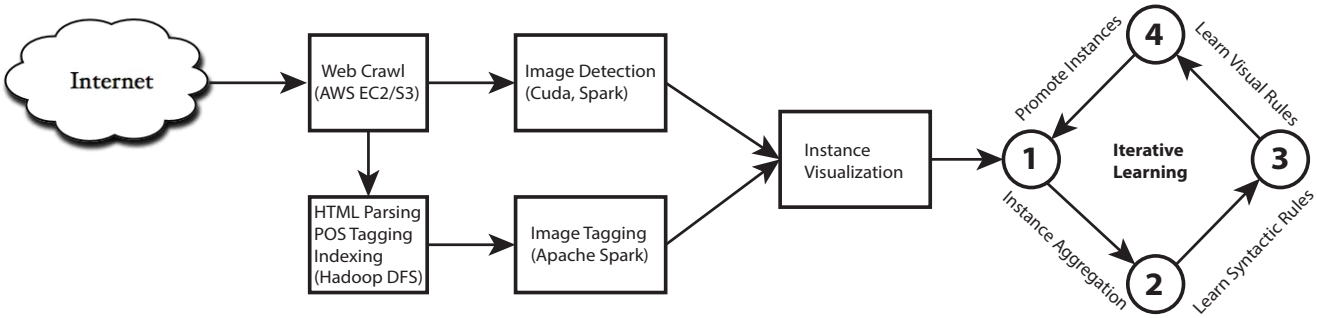

Figure 3: The illustration for information extraction pipeline of our system. The data (text and image) is first collected from the Internet in a distributed manner using multiple independent Amazon EC2 virtual machines. Then for the text data we parse the meta contents, followed by part of speech (POS) tagging on the cleaned text, and finally create necessary index for efficient search/extraction. The resulting data is stored in Hadoop Distributed File System (Hadoop DFS) on a collection of machines for efficient distributed access. After that we assign tags to the images automatically as described in Section 4.1.1. For the image data we apply the trained deep learning models to extract visual objects with GPUs. The instance visualization evaluates related visual concepts for individual noun phrases. When visual concepts of noun phrases are ready, we start iterative multimodal extraction by following steps as described in Algorithm 1.

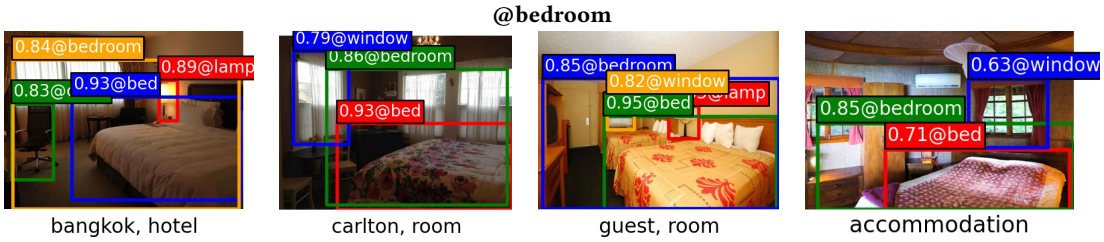

Table 1: Visualization of detected image objects for the bedroom category. For each image, detected objects are marked by rectangles along with confidence values (between 0 and 1) given by the visual object detector. The corresponding text tag of an image is shown at the bottom of that image.

5.2 Multimodal Relations

The multimodal relation analysis automatically discovers related concepts across different modalities. To investigate the effectiveness of the algorithm described in Section 4.1 for learning multimodal relations, we run the program and print the visual relations learned for each category, as shown in Table 2 (only a subset of randomly selected categories are shown due to space limitation). Based on these results, we conclude the following observations:

- For most of the categories, the program is able to discover the correct relations. Occasionally, the program generates some relations that are not consistent with our intuition. For example, *musical instrument* is related to **beach**, etc.
- The program is able to discover some multimodal relations that are difficult to be discovered by human. For example, *pizza* is related to **fish** because many fishes (e.g. tuna, salmon) can be used to make a pizza (or pizza-like sandwich, dishes); *bed* and *kitchen* are related to **airport** because hotels and airports cooccur with high frequency (e.g. "Hilton hotel at O'Hare International Airport").

5.3 Comparative Evaluation

In this section, we compare the capability of extracting information from the Web of the following three different approaches.

- **Proposed:** The multimodal learning algorithm makes use of information across text and image modalities by applying both syntactic rules and visual rules, as described in Section 4.
- **CPL:** It performs information extraction using only syntactic rules only as described in Section 4. The difference between CPL and the proposed approach is that, the proposed approach utilizes visual information in addition to text. It is mostly a re-implementation of the state-of-the-art algorithm called Coupled Pattern Learner (CPL) [6]. The CPL is one of the major extractors for the NELL [5] knowledge base.
- **CPL+Naive Multimodal Fusion (CPL-NMF):** Rather than using the proposed multimodal rules in equation 18, it uses the following scoring function:

$$S(e) = (1 - \prod_{x \in R_t(e)} (1 - P(x))) \cdot \sum_{y' \in \text{IC}} V'(e, y')F(y'),$$

Category	Related Visual Concepts
bridge	bridge, suspension bridge
coach	man's clothing, musician
fish	pizza, striped bass, salmon
clothing	skirt, woman's clothing, jersey
automobilemaker	car, beach wagon, sports car
city	people, sky, window
lake	seashore, ship, cliff, sky
actor	man's clothing, sunglasses
vehicle	warplane, bus, motorcycle, ship
beach	seashore, beach, musical instrument
bird	wading bird, female child, loon
company	computer screen, laptop
hotel	building complex, bed
fruit	orange, fringe tree
airport	warplane, bed, kitchen

Table 2: The related visual concepts learned by multimodal relation analysis

Category	CPL	CPL-NMF	Proposed
vehicle	69.43	80.24	85.75
automobilemaker	86.23	90.11	95.16
fish	80.91	75.67	92.86
bird	68.24	71.28	80.22
bridge	42.57	48.62	52.81
hotel	68.47	78.04	76.45
clothing	80.41	91.72	94.11
airport	85.73	81.37	90.22
musicinstrument	88.14	87.41	91.75
consumerelectronicitem	65.20	70.84	75.23
beach	70.32	71.22	73.04
lake	59.66	63.97	62.90
river	79.64	81.08	89.22
company	96.41	94.54	97.65
plant	74.35	80.22	81.45
insect	71.03	76.25	83.82
city	95.88	91.57	94.47
coach	93.27	95.22	95.74
fruit	74.68	65.32	67.54
actor	95.73	98.65	98.17
athlete	90.05	92.11	94.11
governmentorganization	68.37	70.21	71.43
drug	98.22	96.74	97.83
ceo	79.68	79.24	77.21
Average	**78.44**	**80.48**	**84.13**

Table 3: The comparative results. All the listed categories have number of promoted instances between 94 and 100 for totally 10 iterations.

where $R_t(e)$ is a set of matching syntactic rules and $P(x)$ represents the estimated precision of the rules as before. The IC denotes a set of all visual concepts, and $F(\cdot)$ is the frequency of visual concept for the target category calculated on the promoted samples. The $V'(\cdot)$ is normalized score of V that is defined in equation 6.

The **CPL** is a recent state-of-the-art unimodal approach only using text information, and **CPL-NMF** makes use of multimodal information by naive multimodal cooccurrence.

5.3.1 Experimental Procedure. When comparing the algorithms, we ran each algorithm for 10 iterations of bootstrapping, and then assessed the instances promoted by the algorithms. At each iteration, we promote at most 10 instances and 5 syntactic rules per category by their confidence scores. For each category, we sampled 50 instances to estimate the accuracy of that category. Samples that can be found in the NELL knowledge base [5] are considered as True, and for samples that are not contained in the NELL are manually evaluated by human workers.

5.3.2 Accuracy Comparison. The accuracy comparison results are shown in Table 3. The accuracy of a category is defined as number of correctly promoted instances divided by total number of promoted instances. The "Average" row averages accuracy across all categories. Based on these results, we observe that on average both **CPL-NMF** and **Proposed** outperform the unimodal **CPL** by a quite clear margin. This confirms that multimodal analysis can improves the extraction accuracy over unimodal approach. Additionally, while both **CPL-NMF** and **Proposed** utilize multimodal information, the **Proposed** is able to make more efficient use of the information, which leads to improved accuracy. We also observe that, improvement brought by multimodal analysis is much more significant when target categories are visualizable than those are not. This suggests that, strong relationship between multimodal objects is a crucial requirement for the success of multimodal analysis.

5.3.3 Significance Testing. To test the significance level of the results, we apply the sign test which is a statistical method to test for consistent differences between pairs of observations. According to Table 3, when comparing the **Proposed** and CPL, we see the proposed approach "wins" 20 out of 24 trials. This leads to a p-value (two-tail) of 0.0015, which indicates the improvement over state-of-the-art unimodal approach is significant at the 5% level.

6 CONCLUSIONS

In this paper, we present a novel multimodal analysis approach for text information extraction. The proposed approach consists of three steps, from multimodal relation analysis to learning multimodal rules. The key idea of our approach is to effectively utilize information across multiple modalities for enhanced performance. For the future work, we will study other useful ways to integrate multimodal information to further improve the performance.

7 ACKNOWLEDGEMENTS

This work is supported by DARPA under FA8750-12-2-0348-2 (DEFT / CUBISM), and NSF IIS Award #1526753.

REFERENCES

[1] Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*. ACM, 85–94.

[2] Daniel M Bikel, Richard Schwartz, and Ralph M Weischedel. 1999. An algorithm that learns what's in a name. *Machine learning* 34, 1-3 (1999), 211–231.

[3] Sergey Brin. 1998. Extracting patterns and relations from the world wide web. In *The World Wide Web and Databases*. Springer, 172–183.

[4] Michael J Cafarella, Jayant Madhavan, and Alon Halevy. 2009. Web-scale extraction of structured data. *ACM SIGMOD Record* 37, 4 (2009), 55–61.

[5] 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*, Vol. 5. 3.

[6] Andrew Carlson, Justin Betteridge, Richard C Wang, Estevam R Hruschka Jr, and Tom M Mitchell. 2010. Coupled semi-supervised learning for information extraction. In *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 101–110.

[7] Minmin Chen, Alice Zheng, and Kilian Weinberger. 2013. Fast image tagging. In *Proceedings of the 30th international conference on Machine Learning*. 1274–1282.

[8] James R Curran, Tara Murphy, and Bernhard Scholz. 2007. Minimising semantic drift with mutual exclusion bootstrapping. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, Vol. 3.

[9] Nilesh Dalvi, Ravi Kumar, and Mohamed Soliman. 2011. Automatic wrappers for large scale web extraction. *Proceedings of the VLDB Endowment* 4, 4 (2011), 219–230.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

[11] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 601–610.

[12] Ross Girshick. 2015. Fast R-CNN. In *International Conference on Computer Vision (ICCV)*.

[13] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*. ACM, 675–678.

[14] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, et al. 2015. DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web* 6, 2 (2015), 167–195.

[15] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics* 19, 2 (1993), 313–330.

[16] Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, 188–191.

[17] James Murty. 2008. *Programming amazon web services: S3, EC2, SQS, FPS, and SimpleDB*. " O'Reilly Media, Inc.".

[18] Feng Niu, Ce Zhang, Christopher Ré, and Jude W Shavlik. 2012. DeepDive: Web-scale Knowledge-base Construction using Statistical Learning and Inference. *VLDS* 12 (2012), 25–28.

[19] Marius Paşca, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, and Alpa Jain. 2006. Names and similarities on the web: fact extraction in the fast lane. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 809–816.

[20] Marco Pennacchiotti and Patrick Pantel. 2009. Entity extraction via ensemble semantics. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. Association for Computational Linguistics, 238–247.

[21] Ellen Riloff, Rosie Jones, et al. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI/IAAI*. 474–479.

[22] Benjamin Rosenfeld and Ronen Feldman. 2007. Using corpus statistics on entities to improve semi-supervised relation extraction from the web. In *Annual Meeting-Association for Computational Linguistics*, Vol. 45. 600.

[23] Helmut Schmid. 1995. Treetagger| a language independent part-of-speech tagger. *Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart* 43 (1995), 28.

[24] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, and Robert Chansler. 2010. The hadoop distributed file system. In *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on*. IEEE, 1–10.

[25] Jason R Smith, Herve Saint-Amand, Magdalena Plamada, Philipp Koehn, Chris Callison-Burch, and Adam Lopez. 2013. Dirt Cheap Web-Scale Parallel Text from the Common Crawl.. In *ACL (1)*. 1374–1383.

[26] 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*. ACM, 697–706.

[27] Roman Yangarber. 2003. Counter-training in discovery of semantic patterns. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, 343–350.

[28] Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. 2010. Spark: Cluster Computing with Working Sets. *HotCloud* 10 (2010), 10–10.

[29] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *The Journal of Machine Learning Research* 3 (2003), 1083–1106.