BioSnowball: Automated Population of Wikis *

Xiaojiang Liu† Zaiqing Nie‡ Nenghai Yu † Ji-Rong Wen‡

†MOE-MS KeyLab of MCC
Univ. of Sci. and Tech. of China
Hefei, 230027 P.R. China
xiaojiangliu84@hotmail.com
ynh@ustc.edu.cn

‡Microsoft Research Asia
No. 49 Zhichun Road
Beijing, 100080 P.R. China
{znie,jrwen}@microsoft.com

ABSTRACT

Internet users regularly have the need to find biographies and facts of people of interest. Wikipedia has become the first stop for celebrity biographies and facts. However, Wikipedia can only provide information for celebrities because of its neutral point of view (NPOV) editorial policy. In this paper we propose an integrated bootstrapping framework named BioSnowball to automatically summarize the Web to generate Wikipedia-style pages for any person with a modest web presence. In BioSnowball, biography ranking and fact extraction are performed together in a single integrated training and inference process using Markov Logic Networks (MLNs) as its underlying statistical model. The bootstrapping framework starts with only a small number of seeds and iteratively finds new facts and biographies. As biography paragraphs on the Web are composed of the most important facts, our joint summarization model can improve the accuracy of both fact extraction and biography ranking compared to decoupled methods in the literature. Empirical results on both a small labeled data set and a real Web-scale data set show the effectiveness of BioSnowball. We also empirically show that BioSnowball outperforms the decoupled methods.

Categories and Subject Descriptors

I.5.1 [**Pattern Recognition**]: Models - Statistical

General Terms

Algorithms, Experimentation

Keywords

Summarization, Fact Extraction, Bootstrapping, Markov Logic Networks

*This work was done when Xiaojiang Liu was visiting Microsoft Research Asia.

KDD'10, July 25–28, 2010, Washington, DC, USA.

1. INTRODUCTION

The World Wide Web has been growing rapidly as a huge knowledge repository, containing various kinds of valuable semantic information about real-world entities, such as people, locations, and organizations. Internet users often have the need to find concisely summarized information about different aspects of a person of interest, for example, "When and where was Bill Gates born?", "Who is his spouse?", and "What are the major milestones of his career?". However, current search engines can only return a list of web pages related to such queries, while the information about a single person may appear in thousands of web pages. Therefore, users have to sift through lots of the pages to get a complete view, which is a heavy and tedious job.

In spite of the failure of search engines to return summarized knowledge, Wikipedia enjoys great success in providing knowledge about well-known entities and becomes the first stop for celebrity biographies and facts. Through collaborative editing, Wikipedia builds an entry page for each indexed person. Many of these entry pages contain an infobox summarizing the key facts of the person [21], and a biography portraying the person in more detail. This style of presentation is very effective in describing the life history of a person. A structured infobox provides an express view, and the unstructured biography narrates a more comprehensive story about the person. However, Wikipedia only indexes famous people: celebrities or notable entities. Wikipedia's collaborative editing is based on the *Neutral Point of View* (NPOV) editorial policy, which has been considered as the cornerstone of Wikipedia [6]. However, it is very difficult to reach the NPOV among human contributors on people who are not so notable, for there may be only very little common sense knowledge about the subject which could be collaboratively edited; most of the knowledge is known by few human readers. The NPOV policy has restricted Wikipedia when it comes to providing good summaries for everyday individuals. In this paper, we introduce an automatic approach to summarize the Web to generate Wikipedia-style pages for any person with a modest web presence. As we consider all the public information available on the Web, we believe it will be more trustworthy and manipulation-resistant than knowledge contributed by a few human editors.

1.1 Motivating Example

We have been developing an entity search engine called EntityCube[1]. EntityCube is a different kind of search en-

[1]http://entitycube.research.microsoft.com

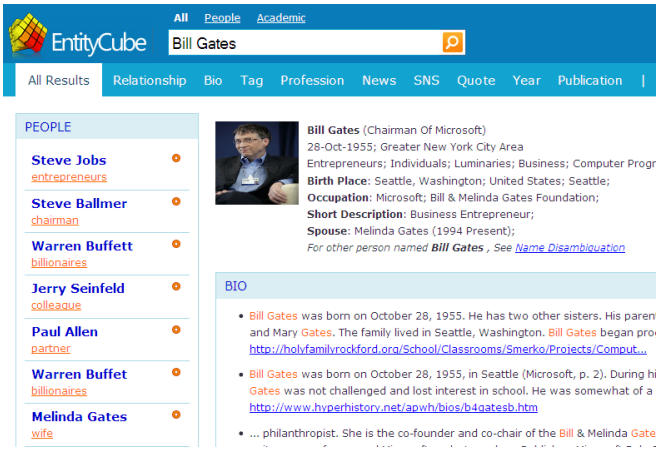

Figure 1: An entity summarization page for the query "Bill Gates" generated by EntityCube.

gine, one that provides summarizations for real-world entities like people, locations and organizations on the Web. In EntityCube, users can ask queries about the entities and explore their relationships. EntityCube is the English-language version of a wildly popular Chinese people search service called Renlifang (we deployed Renlifang last year and introduced it in our WWW 2009 paper [24]). Currently, EntityCube has indexed over 3 billion English web pages. For each crawled web page in EntityCube, the system extracts entities from the page using our Web entity extraction technologies. Knowledge extracted from 3 billion web pages covers a spectrum of everyday individuals and well-known people, locations, and organizations. Compared to Renlifang, EntityCube significantly improved the biography ranking and fact extraction technology. For each person name extracted from the Web, EntityCube tries to automatically generate summary pages, which contain both the biographical texts and facts, and hyper-linked relevant entities to enable surfing between entities (just like surfing on the Web). In Fig. 1, we show the EntityCube result page for the query "Bill Gates". Below we list the key features of EntityCube:

People Biography Ranking. EntityCube ranks text blocks (i.e. paragraphs) from web pages by the likelihood of their being biography blocks.

People Fact Extraction. EntityCube extracts key facts about entities from web pages, such as professions, description tags, and relationships.

Entity Relationship Mining and Navigation. EntityCube enables users to explore highly relevant information during searches to discover interesting relationships about entities associated with their queries.

Expertise Finding. EntityCube can return a ranked list of people known for any interested topics, such as dancing, data mining, conditional random fields, etc.

Web-Prominence Ranking. EntityCube detects the popularity of an entity and enables users to browse entities in different categories ranked by their prominence on the Web during a given time period.

To automatically generate Wikipedia-style people summaries over large scale Web data, we face two challenges: biography ranking and fact extraction. While there are many previous works on each task, a few recent works jointly consider these two types of information. [19] uses facts to rank biography, and [12] addresses the problem of extracting biographical facts. But these works solve the tasks independently, either assuming one type of information is given, or sequentially solving them. But from our observation, these two tasks have strong connections and should be performed together within a single integrated process. On one hand, a good biography always contains many key facts, and these facts are narrated in certain order using natural languages to make them look like a story. For example, below is the first paragraph from Bill Gates' Wikipedia page[2]:

***William Henry "Bill" Gates** III (born **October 28, 1955**) is an **American** business **magnate**, **philanthropist**, **author**, and **chairman of Microsoft**, the software company he founded with Paul Allen. He is ranked consistently one of the world's **wealthiest people** and the wealthiest overall as of 2009. During his career at **Microsoft**, Gates held the positions of **CEO** and **chief software architect**, and remains the largest individual shareholder with more than 8 percent of the common stock. He has also authored or co-authored several books.*

We have marked the facts about Bill Gates in bold. By reading the above biography, we can at least know the facts about Bill Gates' birth name, birthday, professions, titles, honors, etc. The biography can be considered as a set of integrated facts using natural language texts. From this aspect, we can rank the biographical texts based on how many key facts have been covered by the text block, i.e., the more key facts the better. On the other hand, if a block is a biography block, facts in the block are mostly about the subject person, and restrictions on fact extraction on these blocks could be relaxed. We call this property *Bio-Fact duality*.

Besides, existing works always adopt the supervised learning methods, which require a set of human tagged examples to learn the summarization model. Due to the diversity of facts and biography blocks on the Web, supervised methods are not scalable for Web-scale applications like EntityCube. Bootstrapping methods, which start with only a small number of seeds and iteratively enlarge the knowledge base, have been proven effective on Web-scale learning in many fields [24, 1, 5].

In this paper, we propose an integrated bootstrapping framework called BioSnowball to automatically summarize the Web for any person with a modest Web presence. By adopting the bootstrapping framework, BioSnowball starts with only a small number of seeds and iteratively identifies facts and selects biographies. The joint summarization model in BioSnowball performs fact extraction and biography ranking in a single integrated training and inference process using Markov Logic Networks (MLNs) as its underlying statistical model. As the duality property suggests, a joint summarization model can improve the accuracy of both fact extraction and biography ranking, compared to decoupled methods in the literature. Besides the improvements of extraction, the results of the joint summarization can provide rich information for biography de-duplication and person disambiguation.

To the best of our knowledge, BioSnowball is the first working system that takes a bootstrapping architecture and optimizing fact extraction and biography ranking together

[2]http://en.wikipedia.org/wiki/Bill_Gates

in a unified summarization framework. Specifically, we make the following contributions:

(a) We introduce a bootstrapping framework called BioSnowball to jointly perform fact extraction and biography ranking. Compared to the previous decoupled works, BioSnowball has the following advantages:

  i BioSnowball adopts the bootstrapping framework to iteratively find people biographies on the Web. To the best of our knowledge, no previous works use the bootstrapping framework to do the entity summarization, while the bootstrapping framework has been proved efficient on other Web-scale problems [5].

  ii BioSnowball uses the Bio-Fact duality and performs fact extraction and biography ranking within a single integrated training and inference process.

(b) We extensively evaluate BioSnowball and empirically show that BioSnowball can both generate facts with high precision/recall and identify the biography blocks for a wide range of entities.

(c) BioSnowball is efficient and has been evaluated in the context of EntityCube.

In this paper we focus on generating summarization pages for people, however the same technologies could be easily adapted to solve the summarization problems for other types of entities such as organizations and products. This is because the description of a non-person entity is just like a biography of a person, and the description will include many key facts of the entity as well. We can still use the BioSnowball framework to perform description ranking and fact extraction together.

The rest of the paper is structured as follows. Section 2 formally defines the summarization problem. Section 3 gives a brief overview of the BioSnowball system. Section 4 presents the joint summarization model and the training and inference of the model. Section 5 presents our empirical results. Section 6 discusses some related work, and Section 7 concludes this paper.

2. PROBLEM FORMULATION

As discussed in the previous section, we target at building an automatic summarization framework to generate a summary page with key facts and biography blocks for any person with a modest Web presence. In this section, we will formally define the problem and introduce the notations used in the paper.

2.1 Web Blocks

A person may appear in thousands of web pages. However, for most cases, only a small region of a web page contains descriptive information. For example, shown in Fig. 2, there are only three regions (labeled as Web block 1, 2, and 3) containing descriptive information. We call these semantically coherent data regions of a web page *Web blocks* (or *blocks*). In EntityCube, we employ the VIPS algorithm [25] to segment web pages into blocks. Among the different types of blocks generated by the VIPS algorithm, only the information blocks displayed in the center of the page are used. The block is more semantically coherent in nature, and is the basic content unit for our task.

Figure 2: Three Web blocks with text content detected by VIPS on a webpage

Existing works on document summarization and biography generation usually choose sentences as the basic content units, either using the existing sentences extracted from documents [9] or automatically creating them [19]. However, for Web-scale people summarization, sentences are not appropriate any longer, due to the following four reasons. Firstly, when people describe an entity, they do not always mention the entity name in every sentence. Simply collecting the sentences that contain the entity name would miss lots of valuable information. For example, as we can see in Fig. 2, if we select sentences rather than blocks as content units for summarization, most of the information in Web block 2 will be discarded. That is because only the first sentence mentions the name "Jackie Chan". Secondly, sentences are usually too short to fully describe one specific aspect of an entity. Thirdly, it is difficult to obtain coherent summaries by combining sentences extracted from different web pages [4]. Finally, the Web is a huge information repository that normally contains enough informative blocks for people information summarization, thus there's no need to complete sentence extraction. The user study in the context of EntityCube shows that BioSnowball can summarize a quite large range of people with only a modest Web presence.

EntityCube has built a name-to-block inverted index which maps each entity name to Web blocks that contain it. We call these Web blocks the *Contextual Blocks* $\mathcal{B}_e$ of entity e.

2.2 Facts and Biographies

We generate two types of summaries for each person: key facts and biography blocks.

Facts: A fact can be considered as a ternary tuple (e, p, v), where e is an entity, p is the property tag or fact type, and v is the value. Based on our observation, most values of entity facts are noun phrases or numbers, for example, birth places, profession or tags. Thus in this paper, we define the fact to be a binary relation between the subject entity and a noun phrase or a number (all refer to noun phrase), while the relation type is the fact type p. A fact can be represented as (e, np, key), where np is a noun phrase and key is the keyword that indicates the fact type.

Biographies: A biography is a description about the subject entity's life, which presents the subject's story, high-

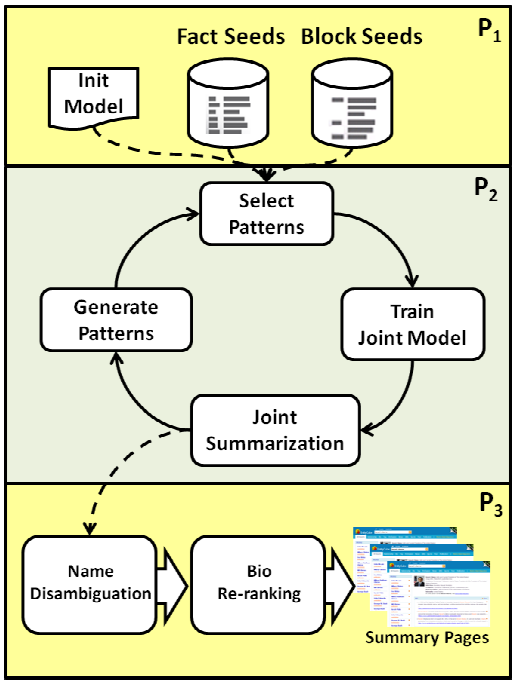

Figure 3: The architecture of BioSnowball

lighting various aspects of his or her life, including intimate details of experiences, and may include an analysis of the subject's personality[3]. As we use blocks as substitution for paragraphs, a biography can be represented as $\{b_1, b_2, ..., b_l\}$, where $b_i \in \mathcal{B}_e$ and is a *biography block* (*bio-block* for short) which describes some aspects of the subject entity e's life. l is a pre-defined threshold to restrict the return block count.

2.3 Joint Summarization Task

To best summarize the information of a person on the Web, we face the problem of joint summarization of facts and biographies. Below we formally define the Joint Summarization task.

Definition 1. **(Joint Summarization of Facts and Biographies)**: Given a specific person e and his/her contextual blocks $\mathcal{B}_e$, the joint summarization problem is to jointly find the top-l non-redundant bio-blocks within $\mathcal{B}_e$ and non-redundant facts extracted from $\mathcal{B}_e$ to best describe the person.

3. OVERVIEW OF BIOSNOWBALL

In this section, we will give a brief overview of the bootstrapping framework of BioSnowball. BioSnowball adopts the bootstrapping framework to iteratively extract facts and select bio-blocks. While starting with a small number of seeds, BioSnowball can automatically identify both the facts and bio-blocks, and refine the summarization model. Fig. 3 shows the architecture of BioSnowball. Generally, BioSnowball has three parts: Input (P_1), Bootstrapping Model (P_2), and Post-Processing (P_3). The Input part provides initial seeds and training data sets for the bootstrapping joint summarization model or an initial summarization model. Using the input from P_1, P_2 iteratively trains the joint summarization model and infers the summaries. In P_3, post-processing techniques, such as name disambiguation and biography de-duplication, are applied to the output of the summarization model to distinguish different people with the same name and make the biography more diverse.

[3]http://en.wikipedia.org/wiki/Biography

3.1 P_1: Input

The input part P_1 contains a set of initial seeds, including the fact seeds and seed blocks or an initial summarization model. The seed blocks are not restricted to be bio-blocks; blocks containing seed facts can also be used as the seeds. These blocks are helpful in the fact extraction training and can be considered as unlabeled data to the query "is the block a bio-block?" during the training. The seeds can be collected by parsing Wikipedia pages or directly using the DBpedia database. If an initial summarization model is provided, we could first use this model to do summarization and consider the results as the seeds for the next part.

3.2 P_2: Bootstrapping Summarization Model

The second part P_2 is a bootstrapping summarization procedure. To start, P_2 takes the seeds from P_1 to learn a joint summarization model. The joint summarization model takes the blocks and facts as the input, and performs the inference of fact extraction and bio-block selection together. We do such joint summarization on all the training blocks. For each block, we try to extract facts and classify whether it is a bio-block. Bio-blocks and corresponding facts are then added into the database for the next round training. In a typical bootstrapping process, we need to add some new training data to the whole training data set at the end of every iteration to update the model. Take the bootstrapping relation extraction as an example [1], after we have identified some new relation tuples, we then locate these tuples in other training blocks. According to the pattern-relation duality which states good tuples generate good patterns, we claim good extractions in these blocks, although we do not actually "extract" from these blocks. This will enlarge the training data set and introduce more patterns in the next iteration. Following this idea, we try to locate facts in the training blocks using the keywords and value matching. Blocks with at least one fact are then added into the training data set. These blocks are tagged as unknown to the bio-block query in the training model. Although there is no direct connection between the unknown blocks and bio-block training, through the joint training of fact extraction and biography ranking, this information will indirectly affect the biography ranking.

After getting more training blocks, BioSnowball generates patterns from the training data and applies ℓ_1-norm pattern selection to select most useful patterns. These patterns, together with the training data, are then returned to the first step to re-train the joint summarization model. BioSnowball iteratively performs these four steps (i.e. train joint model, joint summarization, generate patterns, and select patterns) until no new bio-blocks or facts are identified or no new patterns are generated.

3.3 P_3: Post-processing

The third part P_3 ends the summarization with post-processing. The output of P_2 is a set of extracted facts and bio-blocks. For every identified fact and bio-block, we get a rank value which is the conditional probability based on the observation in the joint summarization model. The re-

sults may mix the information about several people with the same name (for some popular names). By using the output from P_2, it will be easier to disambiguate the results with the facts of the person, as suggested in [8]. Several methods, such as single pass clustering, can be applied to all facts and blocks. In the experiments, we have shown an example of how it can be applied.

There may be many duplicated bio-blocks and facts in the results, as we rank the bio-blocks independently in P_2, without knowing other blocks. End users always want to get as much information as possible in a limited time, thus duplicate or near duplicate bio-blocks are a waste of users' time.

To make the summary more diverse, we re-rank these bio-blocks using the MMR algorithm (Maximal Marginal Relevance [7]),which has been widely used in removing the redundancy in information retrieval. As we have already extracted the facts in bio-blocks, the similarity function in MMR can be modified to emphasize the importance of the facts. We define the similarity of two blocks by the overlap of the facts they contain:

$$Sim(b, b') = \sum_{f \in F_b \cap F_{b'}} p(f|e) \quad (1)$$

where F_b is the set of facts extracted in bio-block b and $p(f|e)$ is the probability that fact f of e is true. The similarity of bio-block b and b' is the total sum of the overlapped fact ranks of entity e.

4. JOINT SUMMARIZATION MODEL

In this section, we introduce the joint summarization model, which jointly performs the two tasks of people summarization: fact extraction and biography ranking. A biography is composed of key facts of the entity and can be ranked by the facts it contains. Facts about the subject entity can be more easily extracted from the biography than non-biographies. We adopt the general discriminative statistical model Markov Logic Networks (MLNs) as the underlying extraction model to capture such dependency. The reason is the following: first, MLN is a discriminative model that can incorporate any features without strong independence assumptions as made in generative models. By using general patterns, we can get better coverage and perform various kinds of extraction [24]. Second, joint inference can be conveniently achieved by adding knowledge as formulae into the MLN.

We first briefly introduce the MLN model, and then these two tasks are separately considered using MLN, which also serves as part of the joint model. We add formulae to connect them together to enable joint inference. Lastly, we consider the training and inference of the joint model.

4.1 Markov Logic Networks

Recently, many NLP applications leverage statistical relational learning and structured prediction which can greatly improve the performance. A Markov Logic Networks is a probabilistic extension of first-order logic and softens the hard constraints by assigning a weight to each formula. The weight indicates how strong the corresponding formula is. When a world violates one formula in the knowledge base it is deemed less probable, but not impossible.

An MLN can be considered as a template to construct a Markov network. To obtain the Markov network, MLN creates binary nodes for all possible groundings of each predicate and creates edges between two nodes if and only if the corresponding ground predicates appear together in at least one grounding of one formula. In our tasks, we get two fixed query predicates and many evidence predicates. Thus, we partition the ground atoms into two sets—the set of evidence atoms X and two query atoms Q, and define a discriminative MLN [16]. Discriminative models have shown great promise compared to generative models in many applications [14, 16]. In BioSnowball, X can be all the possible features we automatically generate from the inputs, and Q are the fact queries $Fact$ and bio-block queries $BioBlock$. Given an input $\mathbf{x}$ (e.g., blocks, seeds and their features), the discriminative MLN defines a conditional distribution $p(\mathbf{q}|\mathbf{x})$ as follows:

$$p(\mathbf{q}|\mathbf{x}) = \frac{1}{Z(\mathbf{w}, \mathbf{x})} \exp\Big(\sum_{i \in F_Q} \sum_{j \in G_i} w_i g_j(\mathbf{q}, \mathbf{x}) \Big), \quad (2)$$

where F_Q is the set of formulae with at least one grounding involving a query atom, G_i is the set of ground formulae of the ith first-order formula, and $Z(\mathbf{w}, \mathbf{x})$ is a normalization factor, also known as partition function in physics. $g_j(\mathbf{q}, \mathbf{x})$ is a binary function and equals to 1 if the jth ground formula is true and 0 otherwise. By using the first-order formulae to construct Markov networks, MLNs have the power to concisely specify very complex relation models, and are ideally suited for joint inference problems. Several tasks [17, 15] have successfully performed the joint inference using MLNs.

Table 1: Relation Type Definition

Type	Sample Keywords	Value Type
birth	born, birth	date, location
death	dead, death	date, location
profession	professor, scientist	organization
known as	known, well known	people
related people	wife, successor	people
work	compose, paint, found	noun phrase
education	graduate, get degree	organization

4.2 Fact Extraction

As defined in the previous sections, a fact in BioSnowball can be considered as a ternary tuple (e_i, np_j, key), where e_i is an entity, np_j is a noun phrase, and key is a set of keywords that indicate the fact type. As in most of the fact extraction systems, we assume that the entities and the noun phrase chunking results are given and only focus on deciding whether the fact is true. To ensure high precision, we restrict the fact types to be extracted in a pre-defined set which contains the most important biographic fact types for the summarization. In this paper, we use 7 such types: birth, death, profession, known as, related people, work and education. These 7 types are selected from the most frequent fact types occurring in Wikipedia. Table 1 shows the detailed definition of these types.

We define the query predicate as $Fact(e_i, np_j, b_k, r_l)$, where b_k is a block containing the entity e_i and noun phrase np_j, and r_l is a fact type. For each combination of e_i, np_j, b_k, r_l, MLN infers a probability for the query $Fact$, which can be used as the confidence of the fact between entity e_i and noun phrase np_j in block b_k of fact type r_l. By using the discriminative model, we can define arbitrary features, including POS tag sequence, distance between the entity and noun

Table 2: Keywords Matching Patterns

Types	Requirements	Example
VB	not stop word and occur more than MIN_OCCUR times	(e_1, *marry*, e_2)
IN	if the token appear more than MIN_OCCUR times and the previous token is a noun phrase	(e_1, CEO *of*, e_2)
POS	the following token is a noun phrase	(e_1, *'s* career, e_2)

phrase, etc. To achieve good balance between the specificity and coverage [5], two kinds of patterns are used:

(a) General patterns: We introduce many general patterns for the extraction process, mainly based on the POS tagging results, which have been proved to be effective in the information extraction task. All the blocks in the data set are parsed using a POS tagger and an NP chunker. The POS tag sequences appearing between the entity and the noun phrase are used as general POS patterns. Some general features about the candidate noun phrase can also help to decide the relationship, such as texture features of the noun phrase (e.g. noun phrases ending with "university" may indicate relationship "alma mater").

(b) Keyword-matching patterns: Keyword-matching patterns generally have high specificity and low coverage. In BioSnowball, we use the keyword matching to balance the flexibility introduced by general patterns. Apart from the keywords pre-defined in the fact types, we automatically discover keywords in the pattern generation process. Only the keywords that may be ambiguous expressed with POS tags are considered. Table 2 shows the detailed configuration.

These patterns are used to form the formula in MLN with the following template:

$$Pattern(e, np, b, +r) => Fact(e, np, b, +r),$$

where $Pattern$ stands for all the possible patterns and "+r" notation signifies that the MLN contains an instance of this rule for each fact type.

4.3 Biography Ranking

A biography tends to have relatively fixed writing styles and specific vocabulary, such as "is born", "graduated from", etc. Similar to the soft pattern matching used in [10], we use the context window around the person name as the biography patterns. For every name in the block, a pre-defined window of size w is used to get left and right context words around the entity, and automatically generate biography patterns. Similar to fact extraction, these patterns form the formula in MLN with the following template:

$$BioPattern(e, b) => BioBlock(e, b),$$

where $BioPattern$ stands for all the possible biography patterns, $BioBlock$ is also a query predicate which means block b is a bio-block with the subject entity e. Besides the context window patterns, general features, such as whether the entity is at the beginning of the block, or whether the word "I" or "you" occurs in the block, can also be introduced.

4.4 Joint Summarization

Fact extraction and biography ranking can help each other. Below we consider three cases:

(a) As the Bio-Fact duality indicates, biography can be ranked by facts it contains. We add the following formula to enable such dependency:

$$Fact(e, np, b, +r) => BioBlock(e, b),$$

(b) Biographies always first mention the subject entity with the whole entity name, and will refer to the target person using pronouns or short names. This is an active research topic [15] and commonly known as the co-reference resolution problem. The co-reference problems raise the difficulty of information extraction because we do not know whom the pronoun refers to. But if we have already inferred that a block is a bio-block, the co-reference problem becomes easier because most of the pronouns are the subject entity. To verify the assumption, we randomly select 20 person's biographies from Wikipedia and Biography.com[4], and manually evaluate whether the pronouns refer to the subject entity. Only 2 cases (out of 630) do not refer to the subject entity. To incorporate such dependency, we add a formula inferring from $BioBlock$ to $Fact$:

$$BioBlock(e, b) \wedge CoRef(e, pr) \wedge Pattern(pr, np, b, +r) \Rightarrow Fact(e, np, b, +r),$$

where $CoRef$ means a co-reference between the entity e and a pronoun pr.

(c) Biographies always follow some traditional styles. A biography normally starts with the subject name and the birth date, birth place, and then death information if the subject entity has passed away. After that, life experience or personal achievements are presented in chronological order, such as education, scholarship, marriage, retirement and etc. These dependencies inspire us to model the fact dependencies in bio-blocks: if a fact f of type r is found in a bio-block, some fact types are more likely to appear after or before that. We add the noun phrase relative position evidence predicate $Next(np, np')$, which means that np' is the next noun phrase after np. A formula is added to model this kind of sequential dependency:

$$BioBlock(e, b) \wedge Fact(e, np, b, +r) \wedge Next(np, np') \Rightarrow Fact(e, np', b, +r'),$$

$+r$ and $+r'$ mean BioSnowball will train this kind of dependency for all fact type combinations.

[4]http://www.biography.com

4.5 Training and Inference

In BioSnowball, there are two query predicates: $Fact$ and $BioBlock$. While the positive training data can be obtained either from the seeds or extracted by previous iterations, we automatically generate the negative training data set. In the training data, the $Fact$ information is known and labeled, but parts of training data are unknown to the query $BioBlock$. In the bootstrapping process, we use existing fact tuples to find more blocks, without knowing if they are bio-blocks. These blocks are added to the training data with the unknown $BioBlock$ tag. The weight learning in BioSnowball is a mixture of the supervised and unsupervised learning [15], which both target at maximizing the conditional log-likelihood

$$\begin{aligned} L(\mathbf{x}, \mathbf{q}) &= log\ p(Q = \mathbf{q} | X = \mathbf{x}) \\ &= log\ p(F = f, B = b | X = \mathbf{x}), \end{aligned}$$

where Q are the query predicates: F the query $Fact$ and B the query $BioBlock$. But for the part of unsupervised learning, we sum the probability over the unknown query predicate B, and maximize the conditional log-likelihood

$$L(\mathbf{x}, \mathbf{q}) = log \sum_b P(F = f, B = b | X = \mathbf{x}),$$

The Markov logic software package Alchemy[5] provides such supports by adding "?" before the unknown query predicate. For those unknown training blocks, we add the grounding $?BioBlock(e, b)$ in the training data set.

The training and inference time mainly depend on the complexity of the model and the number of queries. The complexity of BioSnowball largely depends on the number of patterns we automatically generate. To reduce the number of patterns, we complete the ℓ_1-norm regularized maximum likelihood estimation [24] to remove the zero weight fact patterns and biography patterns. Specifically, we first use the generated patterns to formulate a set of candidate formulae of MLN. Here only the non-recursive definite patterns (such as the POS tag sequence patterns) are used. Then, we apply the algorithm [2] to optimize the ℓ_1-norm penalized conditional likelihood function, which yields a sparse model by setting some formulae's weights to zeros. The zero-weighted formulae are discarded and the resultant model is passed to the Markov logic for weight training.

5. EXPERIMENTS

In this section, we report empirical results of BioSnowball with different configurations. We first try to verify the Bio-Fact duality assumption on the Wikipedia data set. We compare the joint summarization model with both the non-joint model and the model without ℓ_1-norm pattern selection, and show the advantages of joint summarization and ℓ_1-norm pattern selection. We show the bootstrapping performance of BioSnowball on the real Web data set. A user study is conducted on the summarization results on different levels of Web presence to show our model can handle quite a large range of people. Finally, BioSnowball is efficient and has been evaluated in the context of EntityCube.

5.1 Data Set

Our experiments use two data sets: the initial seeds and the Web blocks. The initial seeds are obtained from Wikipedia. We crawl Wikipedia web pages using a celebrity list and parse the bio-blocks and infoboxes on the page. About 17850 people with both the infoboxes and bio-blocks have been collected. As stated in [21], the Wikipedia infobox contains much noise. We filter out noise by using a threshold of frequency of the fact types. We set the threshold to 100. Table 3 shows the top 10 frequent fact types. We will refer to this data set as $WikiSeed$.

Table 3: Top 10 Most Frequent Fact Types

Property	Occurrence	Hit in Bio	Hit Ratio
birthdate	**16959**	**16529**	**0.975**
name	**15539**	**12557**	**0.808**
birthplace	**13762**	**13142**	**0.955**
spouse	10888	5471	0.502
occupation	**8022**	**6553**	**0.817**
birthname	**6897**	**5524**	**0.801**
deathdate	**6272**	**6128**	**0.977**
deathplace	**5319**	**4918**	**0.925**
location	5013	3624	0.723
alma mater	**3822**	**3145**	**0.823**

Besides the initial seeds, we collect web pages indexed by EntityCube. We first partition the web pages into blocks using a visual parser [25] and only the blocks in the center of a web page are selected to compose our data set. All the text sentences in the blocks are parsed using a POS tagger to get the POS tagging results. We collect 1 million such blocks and will use these as the training blocks during the bootstrapping process; this data set will be referred to as $Web1M$.

5.2 Bio-Fact Duality

The bootstrapping framework and joint inference model are based on the Bio-Fact duality assumption, which states that given a good set of facts, we can rank a good set of bio-blocks, and the converse statement, given a good set of bio-blocks, we can easily get a good set of facts about the subject entity. In this experiment, we will empirically verify the assumption using the real Wikipedia data. We try to locate facts in the corresponding bio-blocks using both the hyperlink matching and text matching, as described in [21]. To measure how likely a property appears in the biography, we define the hit-ratio to be the number of hits in bio-blocks divided by the total number of occurrences. In total, among the 319338 facts, 206346 facts are found in the biography blocks (64.6%). Table 3 shows the top 10 most frequent fact types, while the bold types get over 80% hit-ratio.

Among the top 10 most frequent fact types, 84.2% occur in the bio-blocks. We count the facts in the bio-blocks with respect to the block position. We found that the first block contains 5.98 facts per person on average (while the average sentence count of the first bio-block is 2.34), the second contains 2.74 facts, and the 3rd to 8th block contains about 1 fact(s) (notice that a fact may occur in many blocks). This signifies that bio-blocks contain many facts, and the more important bio-blocks contain more facts. From these statistics, we can get the following conclusions:

(a) Important facts occur in the biography blocks.

(b) The bio-blocks contain many facts.

(c) The more important bio-blocks contain more facts (the first bio-block contains 2.6 facts per sentence).

[5]http://www.cs.washington.edu/ai/alchemy

5.3 Joint Summarization Model

5.3.1 Methodology

We empirically evaluate the joint summarization model with ℓ_1-norm pattern selection on the Wikipedia data set. To evaluate the contribution of the joint summarization and the ℓ_1-norm pattern selection, we configure 3 different models.

bnBioSnowball: the baseline of our experiments, performs the biography ranking and fact extraction separately.

jnBioSnowball: add joint inference formulae in Section 4.4 to the *bnBioSnowball* model, without pattern selection.

jpBioSnowball: enable ℓ_1-norm pattern selection in *jnBioSnowball*, the joint summarization model with ℓ_1-norm pattern selection.

We randomly select 300 bio-blocks in the $WikiSeeds$ data set as the training data, and select 100 blocks in $WikiSeeds$ and 100 blocks in the $Web1M$ as the testing data set. For the $Web1M$ 100 blocks, we manually label the facts in the block and whether the block is a bio-block. The ℓ_1-norm pattern selection λ has been set to 0.5. Traditional information extraction evaluation criteria precision, recall and F1 are used to evaluate the performance of these systems.

5.3.2 Results

Table 4 shows the evaluation results on the data set. From the results, we can see that jpBioSnowball generally achieves better performance on both biography ranking and fact extraction. From that we can see that jointly training the two tasks can improve the performance by getting more facts and more bio-blocks. Also, we can see that the methods without ℓ_1-norm pattern selection perform worse than the methods with pattern selection. This is due to the pattern selection which removes many uncommon patterns, while the training is much better when there is little noise in the model. Overall, the joint model extracts more facts and bio-blocks and models with pattern selection perform much better.

5.4 Bootstrapping Framework

We empirically configure BioSnowball to complete the bootstrapping framework on the $Web1M$ data set. 1000 bio-blocks from $WikiSeed$ are randomly selected as the initial seeds to start the bootstrapping process. For comparison, we configure BioSnowball without joint inference on facts and bio-blocks, and refer to *basicBioSnowball*. We do not complete experiments using BioSnowball without ℓ_1-norm pattern selection, for they are too slow and in the last experiment we have already shown the effectiveness of pattern selection. The ℓ_1-norm pattern selection λ has been set to 0.5. Since labeling all the facts and bio-blocks on the $Web1M$ data set is impractical, it is difficult to quantitatively evaluate as in the previous experiment. For all the results, we invite three interns in the lab to manually label all the result blocks whether or not it is a bio-block, and randomly select 100 facts from all the extractions to evaluate the precision and good extraction count. Fig. 4 shows the number of correct fact extractions and the precision of the identified facts with respect to the number of iterations. Fig. 5 shows the the number of correct bio-block results and the precision with respect to the number of iterations.

From the results, the bootstrapping framework can iteratively find more facts and more bio-blocks. We can see that BioSnowball gets more correct fact extractions and higher precision than *basicBioSnowball*. *basicBioSnowball* gets a much deeper decrease in precision with respect to the iteration numbers, while BioSnowball performs more stably. The biography ranking achieves similar results, while BioSnowball outperforms *basicBioSnowball* in both the good facts count and the precision. The results on the real Web data set $Web1M$ show that the bootstrapping framework can get more summaries iteratively without sacrificing the precision. The experiment also shows that the joint summarization model performs better than the decoupled model.

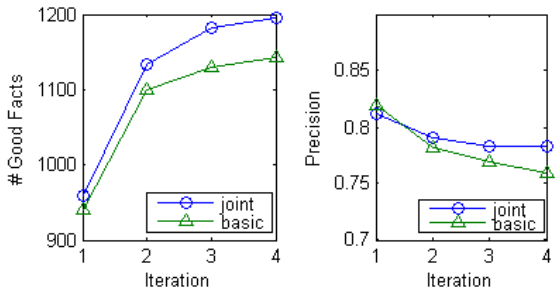

Figure 4: The performance (precision, # good facts) of the two BioSnowball systems during the iteration.

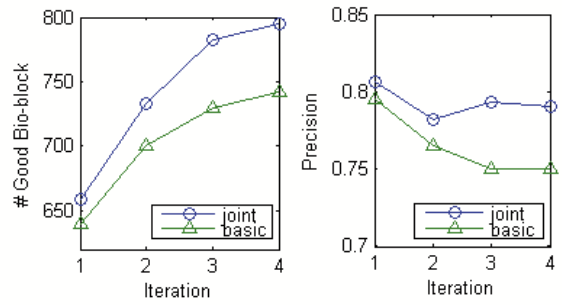

Figure 5: The performance (precision, # good bio-blocks) of the two BioSnowball systems during the iteration.

5.5 Different Levels of Web Presence

We run a user study experiment to show the performance of joint summarization on people with different levels of Web presence. In this experiment, users are asked to rank between summaries generated by BioSnowball and the baseline.

5.5.1 Methodology

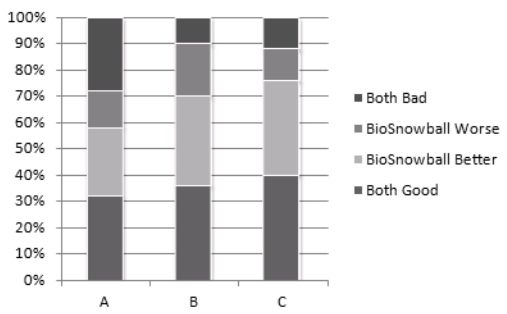

Figure 6: User study results on the summarization in different levels of Web presence

Table 4: Evaluation of fact extraction results of different joint summarization models

Types		Bio	Birth	Death	Overall Facts
bnBioSnowball	*Precision*	0.758	0.951	0.619	0.714
	Recall	0.675	0.791	0.325	0.754
	F1	0.714	0.864	0.426	0.734
jnBioSnowball	*Precision*	0.402	0.615	0.742	0.71
	Recall	0.380	0.064	0.442	0.08
	F1	0.390	0.116	0.554	0.144
jpBioSnowball	*Precision*	**0.933**	**0.954**	**0.758**	**0.794**
	Recall	**0.770**	**0.932**	**0.543**	**0.908**
	F1	**0.844**	**0.943**	**0.633**	**0.847**

Table 5: Summarization example of "Kerry Webb"

Person	Fact-Blocks	Facts
Coach	Dr. Kerry Webb is a talented trainer and professional coach. Kerry's passion for creating positive workplaces and high performance teams makes him the consultant for your organization.	Profession: trainer Profession: coach Alma mater: University of Florida
Professor	Kerry Webb, a Dallas Baptist University professor, will lead a "Walk Through the Bible" seminar March 12 at Cedar Heights Church in Cedar Hill.	Profession: Professor Affiliation: Dallas Baptist University
Officer	Kerry Webb, Policy Officer, InTACT, ACT Government, Managing public sectors' websites interface language and elements - do users and the public understand you?	Profession: Policy Officer Affiliation: InTACT

To evaluate how well we can summarize the people with different levels of Web presence, a user study is conducted to evaluate the summarization performance with respect to the level of Web presence. We define the person's level of Web presence to be a range of numbers of the person's contexture blocks indexed by EntityCube. Three levels are used in this experiment: Level A with 10 - 100 blocks indexed, Level B 100 - 1,000, Level C 1,000 - 10,000. For the people who have more than 10,000 blocks, it becomes much easier to get summaries of them, which we do not consider here. For each level, we randomly select 10 people and retrieve all the contextual blocks of the person. We use the baseline method in the previous section and the BioSnowball method to summarize the entity, while we use the models trained in the previous experiment as the initial model. Both the results have been de-duplicated using the MMR algorithm, and only the top 3 ranked blocks have been presented to the end users. 5 users are invited to label the summarization result using a blind A/B test. There are four choices for each test: both are good, A is better than B, B is better than A, both are bad. A summary is good if it can summarize the different aspects of the person, and users can get as much as possible information from the summary.

5.5.2 Results

Fig. 6 shows the results of the users' evaluation on different levels of Web presence. From the results, we can see that BioSnowball performs better than the baseline method in all 3 levels, especially in levels B and C which stand for the person getting 100 - 10,000 blocks. In level B and C, BioSnowball succeeds in over 70% of cases providing good summarization, and 20% better than the separate method. Even in level A, BioSnowball succeeds in 58% of cases and 13% better than the baseline. The ratio of *Both Good* increases when the available information increases, indicating it is easier to get the summary when the person's Web presence increases. As the experiment shows, BioSnowball can summarize a quite large range of people with only a modest Web presence.

To demonstrate that joint summarization can help the name disambiguation, Table 5 shows the extraction and ranking results of "Kerry Webb" who falls in level B. From the example, we can find that we have extracted facts about his professions, alma mater, etc. Using the facts, such as professions, it is easier to separate different people with the name "Kerry Webb". The blocks can then be aligned to fact clusters using clustering methods.

5.6 Efficiency

The BioSnowball is efficient. It takes about 30 minutes to complete extraction on the $Web1M$ corpus with a standard single-core desktop computer.

6. RELATED WORK

Traditionally, fact extraction and biography ranking are studied in two different domains: information extraction and multi-document summarization. The Web-scaled fact extraction problem is generally solved by the bootstrapping framework [1, 3, 24]. Compared to traditional supervised fact extraction methods [13], bootstrapping methods significantly reduce the number of training examples by iteratively discovering new extraction patterns with only a small set of seeds [11, 1]. In BioSnowball, we adopt the bootstrapping framework and extend it to the field of people information summarization using the statistical approaches [24].

Biography generation has been considered as a multi document summarization problem [23], and is solved by sentence extraction [23] or information extraction [19]. All these methods face the coherent problem [4], which is difficult to solve. We avoid this problem on the Web environment by using the block as the basic content unit, and the biography generation problem can be solved by ranking. Several attempts have been made to rank the biographical texts either by patterns [10] or by facts [19]. Some research focusses on leveraging the biographical text to do information extraction [22]. But they still separately solve these two tasks, while BioSnowball considers these two tasks in a single integrated model.

Recently, there has been a rising trend in refining, popu-

lating Wikipedia or leveraging Wikipedia to help information extraction [21, 20, 18]. While Kylin/KOG [21, 20] focus on customizing and optimizing for Wikipedia articles in fact extraction, BioSnowball uses the Wikipedia knowledge as the seeds to gather more facts and biographies on the Web. SOFIE [18] uses Wikipedia knowledge base (YAGO) as an existing ontology, and applies logical reasoning to populating new facts. But SOFIE only focuses on the fact extraction, while BioSnowball jointly considers the biography ranking and fact extraction, and populates both these two summaries.

BioSnowball is an extension to our previous work StatSnowball [24]. StatSnowball introduces a general statistical bootstrapping framework for Web-scaled entity relationship extraction. BioSnowball adopts the general statistical bootstrapping framework of StatSnowball, but extends it to handle the problems with multiple dependent queries.

7. CONCLUSIONS

This paper presents an integrated bootstrapping framework named *BioSnowball* to automatically summarize the Web to generate Wikipedia-style pages for any person with a modest Web presence. By using the Bio-Fact duality, we jointly perform biography ranking and fact extraction in an integrated statistical model. The joint summarization model uses the general relational model—Markov logic networks (MLNs), which can be configured to model complex dependencies between the inputs and the outputs. By adopting a bootstrapping architecture, BioSnowball significantly reduces the number of human-tagged examples and iteratively mines facts and biography blocks. The empirical studies show that BioSnowball is effective and that the joint summarization model performs better than the decoupled methods.

8. ACKNOWLEDGMENTS

The authors Xiaojiang and Nenghai are supported by the Research Fund for the Doctoral Program of Higher Education (20070358040), the National Natural Science Foundation of China (60933013), and the National High Technology Research and Development Program of China (863 No.2008AA01Z117).

9. REFERENCES

[1] E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain-text collections. In *International Conference on Digital Libraries*, 2000.

[2] G. Andrew and J. Gao. Scalable training of l_1-regularized log-linear models. In *ICML*, 2007.

[3] M. Banko, M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *IJCAI*, 2007.

[4] R. Barzilay and K. McKeown. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 2005.

[5] S. Brin. Extracting patterns and relations from the world wide web. In *International Workshop on the Web and Databases*, 1998.

[6] S. L. Bryant, A. Forte, and A. Bruckman. Becoming wikipedian: Transformation of participation in a collaborative online encyclopedia. In *GROUP*, 2005.

[7] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, 1998.

[8] Y. Chen, S. Y. M. Lee, and C.-R. Huang. Polyuhk: A robust information extraction system for web personal names. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, 2009.

[9] M. Collins and Y. Singer. Multi-document summarization by sentence extraction. In *NAACL-ANLP*, 2000.

[10] H. Cui, M.-Y. Kan, and T.-S. Chua. Soft pattern matching models for definitional question answering. *ACM Trans. Inf. Syst.*, 25(2), 2007.

[11] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised named-entity extraction from the Web: An experimental study. *Artificial Intelligence*, 165(1):91–134, 2005.

[12] N. Garera and D. Yarowsky. Structural, Transitive and Latent Models for Biographic Fact Extraction. In *EACL*, 2009.

[13] S. Harabagiu, C. A. Bejan, and P. Morărescu. Shallow semantics for relation extraction. In *IJCAI*, 2005.

[14] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[15] H. Poon and P. Domingos. Joint unsupervised coreference resolution with markov logic. In *EMNLP*, 2008.

[16] P. Singla and P. Domingos. Discriminative training of markov logic networks. In *AAAI*, 2005.

[17] P. Singla and P. Domingos. Entity resolution with markov logic. In *ICDM*, 2006.

[18] F. M. Suchanek, M. Sozio, and G. Weikum. SOFIE: A self-organizing framework for information extraction. In *WWW*, 2009.

[19] M. White and T. Korelsky. Multidocument summarization via information extraction. In *HLT*, 2001.

[20] F. Wu and D. S. Weld. Autonomously semantifying wikipedia. In *CIKM*, 2007.

[21] F. Wu and D. S. Weld. Automatically refining the wikipedia infobox ontology. In *WWW*, 2008.

[22] W.-T. Yih, J. Goodman, L. Vanderwende, and H. Suzuki. Multi-document summarization by maximizing informative content-words. In *IJCAI*, 2007.

[23] L. Zhou, M. Ticrea, and E. Hovy. Multi-document biography summarization. In *EMNLP*, 2004.

[24] J. Zhu, Z. Nie, and X. Liu. Statsnowball: a statistical approach to extracting entity relationships. In *WWW*, 2009.

[25] J. Zhu, Z. Nie, J.-R. Wen, B. Zhang, and W.-Y. Ma. Simultaneous record detection and attribute labeling in Web data extraction. In *SIGKDD*, 2006.