Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Fine-Grained Semantic Conceptualization of FrameNet

Jin-woo Park
POSTECH, Korea, Republic of
jwpark85@postech.edu

Seung-won Hwang
Yonsei University, Korea, Republic of
seungwonh@yonsei.ac.kr

Haixun Wang
Facebook Inc., USA
haixun@gamil.com

Abstract

Understanding verbs is essential for many natural language tasks. To this end, large-scale lexical resources such as FrameNet have been manually constructed to annotate the semantics of verbs (frames) and their arguments (frame elements or FEs) in example sentences. Our goal is to "semantically conceptualize" example sentences by connecting FEs to knowledge base (KB) concepts. For example, connecting `Employer` FE to *company* concept in the KB enables the understanding that any (unseen) company can also be FE examples. However, a naive adoption of existing KB conceptualization technique, focusing on scenarios of conceptualizing a few terms, cannot 1) scale to many FE instances (average of 29.7 instances for all FEs) and 2) leverage interdependence between instances and concepts. We thus propose a scalable k-truss clustering and a Markov Random Field (MRF) model leveraging interdependence between concept-instance, concept-concept, and instance-instance pairs. Our extensive analysis with real-life data validates that our approach improves not only the quality of the identified concepts for FrameNet, but also that of applications such as selectional preference.

1 Introduction

Overview Understanding verbs is critical to understanding text. For example, the verb "employ" is associated with two different semantic classes described as the USING and the EMPLOYING frames:

- USING: [The few states]$_{\texttt{Agent}}$ [employed]$_{\text{LU}}$ [chemical weapons]$_{\texttt{Instrument}}$.
- EMPLOYING: [Hong Kong manufacturers]$_{\texttt{Employer}}$ [employ]$_{\text{LU}}$ [2m workers]$_{\texttt{Employee}}$ [in China]$_{\texttt{Place}}$.

FrameNet (Fillmore, Johnson, and Petruck 2003) is a lexical resource that describes semantic frames, by annotating verbs[1] (*i.e.*, lexical units or LUs) and their arguments (*i.e.*, frame elements or FEs) that evoke frames. Table 1 shows all of the FE instances for the EMPLOYING frame in FrameNet. This resource is critical for human understanding of verbs and for training applications such as semantic parser (Das, Martins, and Smith 2012) or question answering systems (Pizzato and Mollá 2008). Yet, as shown in Table 1, the instances in FrameNet, being just small samples for the FEs, are often insufficient for supporting machine understanding.

We thus focus on "semantically conceptualizing" FE instances. Take the EMPLOYER FE for example. Our goal is to infer that any instance of *company* concept is equally plausible as sample companies mentioned in FrameNet. A closely related well-known task is "selectional preference" of computing "plausibility" scores for arbitrary instances (Resnik 1996). That is, our task is to find selectional preferences based on concepts of a knowledge base (KB). With such conceptualization, we can infer that "Shell hires 100 workers" is plausible as Shell is an instance of *company* concept in the KB.

Related work To enable this inference on unobserved sentences, existing work uses WordNet concept (Tonelli et al. 2012) or topic modeling (Ritter, Mausam, and Etzioni 2010) to relate the unobserved instances to the observed instances. For example, WordNet concepts identified for FEs (Tonelli et al. 2012) can infer plausibility of the unobserved instances if they belong to the same concept from the observed instances. However, WordNet concepts, being restricted to about 2,000 concepts linked with Wikipedia pages (Bryl et al. 2012), are often too coarse for our purpose. For example, the WordNet concept *container*, enumerating generic containers including envelope or wastebasket, is too general for `Container` FE, as it inaccurately predicts wastebasket as a plausibility score of a cooking utensil. More desirably, we may consider using a knowledge base that materializes concepts of finer granularity, *e.g.*, *cooking container* for COOKING_CREATION frame. In addition, a desirable KB should cover sufficient instances for each concept. To argue that WordNet is limited in terms of coverage, we analyzed English Gigaword Fifth Edition (LDC2011T07) consisting of 180 million sentences from English news. In this corpus, WordNet covers only 33% of instances as shown in Table 2. In contrast, automatically harvested KBs such as Probase covers many instances including named entities (*e.g.*, Microsoft and British Airways), which are not covered by WordNet.

Proposed method To overcome concept and instance sparsity of manually-built KB, we utilize on Probase (Wu et al.

[1] Although FrameNet annotated verbs, nouns, adjectives, and adverbs as lexical units, this paper focuses on verbs.

Table 1: Examples of FE instances in the Employing frame from FrameNet examples.

Frame	FE	Instances
EMPLOYING	Employer	british airways, factory, plant, police, housekeeper, bank, firm, executive, company, industry, airline, institute, manufacturer, defendant, woman, paul, mason, man, sister, he, we, i, she, you
	Employee	person, worker, man, staff, specialist, you, gardener, someone, woman, consultant, actor, winning, contractor, stave, architect, graduate, builder, wife, artist, detective, tradesman, westwood, plaintiff, clark, labourers, outside agency, diver, officer, mark thatcher, vet, her, team
	Position	cleaner, guide, assistant, manager, agent, gardener, aircrew

Table 2: The coverage of FrameNet, WordNet, and Probase. Nouns of several dependents such as nsubj, dobj, pobj, and iobj of the sentences from Gigagword corpus were used for this coverage test.

	FrameNet	WordNet	Probase
Coverage	25%	33%	75%

2012)[2], which contains millions of fine-grained concepts automatically harvested from billions of web documents. This gives us the following two benefits observed in prior research:

- Fine-grained conceptualization covers 3 million concepts observed on the Web.
- More than 75% instances of a raw corpus (Table 2) were already observed in Probase, and its instance coverage can grow indefinitely by automatic harvesting from a larger corpus.

A naive baseline would be adopting existing conceptualization techniques on FE instances (Song et al. 2011). Bayes builds on naive Bayes model for finding concepts with the highest posterior for the instances. CL+Bayes extends this model to cluster before conceptualization to address: 1) heterogeneity of concept, *e.g.*, Employer can be both *company* and *person*, and 2) ambiguity of instance, *e.g.*, apple. However, CL+Bayes targets the scenario of conceptualizing a few words, *e.g.*, a multi-word query, and cannot 1) scale for FE instances (average of 29.7 example instances for all FEs in FrameNet) and 2) leverage the interdependence between concept-instance, concept-concept, instance-instance pairs.

In contrast, we propose a new model addressing these challenges. First, while existing clustering builds on clique finding, which is NP-hard and sensitive to missing observations, we propose an efficient clustering that is robust to some missing observations. Second, we extend a Bayesian model with one-way directed graph between concept and instance, $P(e|c)$, to model the interdependence between concept-concept, instance-instance, and concept-instance.

Our experimental evaluations show the accuracy of both conceptualization and selectional preference using real-life datasets.

2 Preliminary

In this section, we introduce two lexical knowledge bases, FrameNet and Probase (Wu et al. 2012).

2.1 FrameNet

FrameNet (Fillmore, Johnson, and Petruck 2003)[3] is a database of frame semantics (Fillmore 1982). It defines lexical units (LUs) as the pairing of words with their meanings (frames). In our running examples, *employ* belongs to two frames: USING and EMPLOYING. In each frame, the verb *employ* has its frame elements (FEs), which can be direct syntactic dependents. The USING frame, for example, contains two FEs, Agent and Instrument, whose relationships are described in example sentences. FrameNet contains more than 190,000 annotated sentences covering about 12,000 LUs and 10,000 FEs.

2.2 Probase

Probase (Wu et al. 2012) is a probabilistic lexical knowledge base of isA relationships that are extracted from billions of web documents using syntactic patterns such as Hearst patterns (Hearst 1992). For example, from "... *mobile company* such as a microsoft ...", we derive the isA relationship *microsoft* isA *mobile company*.

Probase contains isA relationships among 3 million concepts and 40 million instances. Furthermore, each isA relationship is associated with a variety of probabilities, weights and scores, including the **typicality** scores:

- $P(c|e)$ denotes how typical is concept c for instance e (*i.e.*, instance typicality score). It is computed as:

$$P(c|e) = \frac{n(e,c)}{\sum_c n(e,c)} \tag{1}$$

  where $n(e,c)$ is the frequency observed from e isA c in the corpus. Typically, $P(c|e)$ gives higher scores to general concepts than specific concepts. For example, we have $P(food|pasta) = 0.177 > P(italian\ food|pasta) = 0.014$ in Probase.

- $P(e|c)$, denotes how typical is instance e for concept c (*i.e.*, concept typicality score). It is computed as:

$$P(e|c) = \frac{n(e,c)}{\sum_e n(e,c)} \tag{2}$$

  $P(e|c)$ typically gives higher scores to specific concepts than general concepts. For example, we have $P(pasta|italian\ food) = 0.184 > P(pasta|food) = 0.013$ in Probase.

3 Existing Conceptualizations

In this section, we introduce existing methods of conceptualizing terms into Probase concepts and then discuss their limitations.

[2] Dataset publicly available at http://probase.msra.cn/dataset.aspx

[3] We used the FrameNet 1.5 dataset.

3.1 Problem Statement

Conceptualization is to identify representative concepts associated with the given instances. More formally, let $E = \{e_1, e_2, \ldots, e_n\}$ be the given instances, and let $C = \{c_1, c_2, \ldots, c_m\}$ be the concepts from a knowledge base. The goal of conceptualization is to estimate a representative score of $c \in C$ for E.

3.2 Conceptualization Baselines

Probase, with its isA relationships and probabilities, enables the conceptualization of words and phrases. In other words, Probase maps a set of words and phrases to their concepts. We review several conceptualization mechanisms below.

Baseline I: Bayes Consider the following instances for the $Employer$ FE in the EMPLOYING frame: $E_{empr} = \{$british airways, factory, plant, police, housekeeper$\}$. Good representative concepts in Probase would be *company*, *organization*, *manufacturer*, *worker* and *occupation*. .

The Bayes method (Song et al. 2011) performs conceptualization by finding the concept that has the largest posterior probability using a naive Bayes model for the given instances. Formally, Bayes finds the representative concepts as follows:

$$P(c|E) = \frac{P(E|c)P(c)}{P(E)} \propto P(c) \prod_{e \in E} P(e|c) \tag{3}$$

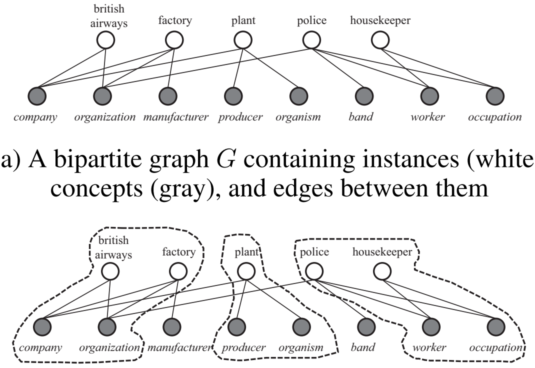
where $P(c)$ is proportional to the observed frequency of concept c in the corpus.

Baseline II: CL+Bayes CL+Bayes extends Bayes to cluster before conceptualization to address: 1) heterogeneity of concept, *e.g.*, EMPLOYER conceptualizes to *company* and *person*, where Bayes identifies too general a concept covering both, *e.g.*, *object*, and 2) ambiguity of instance, such as "plant" for which Bayes conceptualizes to both *company* and *organism* (Fig. 1a).

To address the above problems, CL+Bayes (Song et al. 2011) first abstracts this problem as a bipartite graph where nodes represent instances and concepts, and edges represent *isA* relationships. More specifically, $G = ((U, V), A)$ is a weighted bipartite graph where U is a set of instances, V is a set of concepts, and A is a set of edges. An edge $a(e \in U, c \in V) \in A$ indicates c is a concept of an instance e. They set the edge weight of $a(e, c)$ to the concept typicality score for instance, $P(e|c)$. Fig. 1(a) shows the constructed graph for our running example E_{empr}.

Given the graph, CL+Bayes then clusters semantically-close instances before conceptualization. This process is shown in Fig. 1(b). Term e and concept c is connected by an edge weighted by typicality $P(e|c)$. Given G, CL+Bayes finds the densest k-disjoint cliques (Ames 2012) that maximize edge weights. The instances connected in a clique are treated as a cluster.

Each cluster is then conceptualized by Eq. 3. In our running example, CL+Bayes first divide E_{empr} into three clusters of instances: $\{$british airways, factory$\}$, $\{$plant$\}$, and $\{$police, housekeeper$\}$. Since *worker* and *occupation* are supported by instances such as police and housekeeper in Fig. 1(b), these two concepts form a dense cluster supported by multiple common instances. In contrast, *band* is pruned out, or cannot be a member of clique, as supported only by a single instance.



(a) A bipartite graph G containing instances (white), concepts (gray), and edges between them

(b) G clustered by CL+Bayes.

Figure 1: Overview of conceptualization baselines.

By clustering, CL+Bayes can solve the two limitations of Bayes discussed above: 1) heterogeneity: we cluster semantically-close instances such as $\{$british airways, factory$\}$, $\{$plant$\}$ and $\{$police, housekeeper$\}$; 2) ambiguity: we prune out unrelated concepts such as *band* that will be 0 in probability after clustering.

After clustering, we apply Bayes on each cluster (Eq. 3).

3.3 Limitations

We observe the following limitations for applying this baseline to our target problem:

- **L1: Strict clustering.** Finding the densest k-disjoint clique problem is provably NP-hard (Ames 2012) and also sensitive to some missing observation. Having synonymous concepts (*manufacturer* and *producer*) often leads observations of isA relationship of instances to divide between the two (Fig. 1b). Meanwhile, having a single missing observation between a concept and an instance breaks a clique and affects the clustering.

- **L2: Independence Relationships.** CL+Bayes uses one-way directed graph from concept to instance, $P(e|c)$. Meanwhile, other neighborhood relationships such as concept-concept, instance-instance, instance to concept pairs are not considered.

We thus address these limitations by clustering methods and probability estimation model in Sec. 4.

4 FrameNet Conceptualization

This section proposes a new conceptualization Truss+MRF to overcome the two limitations of the existing work. For **L1**, Sec. 4.1 first proposes a concept synonym smoothing to reduce missing observations and then proposes an efficient and effective clustering method. For **L2**, Sec. 4.2 constructs an undirected probabilistic graph and proposes a new concept probability estimation model to consider relationships between features on the undirected probabilistic graph.

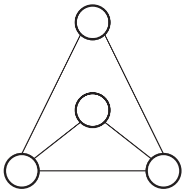


Figure 2: k-Truss when k = 3

4.1 Clustering: Truss

Concept synonym smoothing To overcome $L1$, Bayes implements a classic Laplace smoothing (Lidstone 1920) of adding a small score to missing edges.

$$\overline{P(e|c)} = \frac{P(e|c) + 1}{|U| + |V|} \tag{4}$$

where $|U|$ is the number of instances and $|V|$ is the number of concepts in Probase.

Unlike this concept-agnostic smoothing, we propose concept synonym smoothing to reduce missing observations. We obtain concept synsets from (Li et al. 2013) grouping synonymous Probase concepts into a cluster by k-Medoids clustering algorithm based on the shared instances (instance membership). An example of such concept cluster would be {*job*, *occupation*, *professional*, *worker*}, sharing many common instances. We aggregate observations for all concepts with the same semantics using this concept cluster.

However, although this concept synonym smoothing reduces missing observations, the graph may still have missing observations.

Relaxation of cluster We then propose an efficient approximation of cliques for clustering, with polynomial complexity of $O(n^3)$, which is also more robust to missing observations.

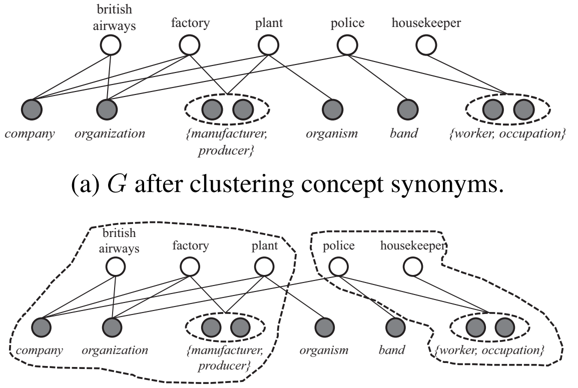
Our algorithm, Truss, finds k-truss, a relaxation of a clique, motivated by a natural observation of social cohesion (Cohen 2008) that two nodes connected by an edge should also share many common neighbors. In a k-truss, an edge is legitimate only if supported by at least k-2 common neighbors of two end vertices. Formally, a k-truss is a one-component subgraph such that each edge is supported by at least k-2 common neighbors. For example, the 3-trusses of a graph include all edges contained in at least one triangle (Fig. 2). With this property of k-truss, incomplete subgraphs including several missing edges can be identified.

Since G is a bipartite graph between instance and concept (Fig. 1a), we cannot apply k-trusses on G immediately. We thus take a two-phase approach. First, we enrich G into $G*$ by creating a link between concepts based on the k-truss intuition. Second, we then identify k heterogeneous concept clusters by running a k-truss algorithm on $G*$.

For the first phase, we connect two concepts c_i and c_j if they have many common instance neighbors. We denote instance neighbors of concept c as $N^I(c)$ such that common instance neighbors are denoted as $N^I(c_i) \cap N^I(c_i)$. Formally, an edge $a(c_i, c_j)$ is inserted when $|N^I(c_1) \cap N^I(c_2)| \geq k_I - 2$ where $N^I(c)$ is the set of instance neighbors linked to the edge. For example, from Fig. 3(a) when $k_I = 4$, {*company*} and {*organization*} are linked to each other because they have two common instance neighbors (*i.e.*, $|N^I(c_1) \cap N^I(c_2)| \geq 2$. In addition, concept vertices c, whose $N^I(c)$ is less than $k_I - 2$, are removed to eliminate ambiguous concepts (*e.g.*, *organism* and *band*) supported by few instances.

Algorithm 1: Truss ($G*$)

```
input : A graph $G*$ consisting of a set of instances $U = \{e_1, \cdots, e_n\}$, a set
        of concepts $V = \{c_1, \cdots, c_m\}$, and a set of edges
        $A = \{a_1, \cdots, a_l\}$
while isChanged() do
    for each edge $a(c_i, c_j) \in A$ do
        if $(|N^C(c_i) \cap N^C(c_j)| < k_C - 2)$ then
            remove $a(c_i, c_j)$;
```



(a) G after clustering concept synonyms.

(b) G clustered by Truss+MRF.

Figure 3: Overview of our conceptualization graph.

For the second phase, Alg. 1 describes how to extract k-trusses for concept nodes from $G*$. First, an edge $a(c_i, c_j)$ is removed when $|N^C(c_1) \cap N^C(c_2)| \leq k_C - 2$ where $N^C(c)$ is the set of concept neighbors. After this process, we identify connected components of concept nodes. Each such concept component (and connected instances) form a cluster, which will be conceptualized into KB concepts.

From our running example, we identify two clusters (*i.e.*, trusses) as shown in Fig. 3(b): {british airways, factory, plant, *company*, *organization*, *manufacturer*, *producer*} and {police, housekeeper, *worker*, *occupation*}. Each concept in clusters is supported by at least two instance neighbors (when $k_I = k_C = 4$). Otherwise, the remaining concepts, which are not supported by at least two instance neighbors, are removed (*e.g.*, *organism* and *band*).

4.2 Posterior Probability Estimation using Markov Random Field: MRF

To overcome **L2**, we use an MRF model (undirected graphical model) to model neighborhood relationships such as concept-instance, concept-concepts, instance-instance pairs. For this purpose, an undirected probabilistic graph is constructed for each cluster. We first propose a new edge weighting on

Table 3: Examples of top-4 concepts ranked by $P(c|e)$, $P(e|c)$, and ours (*i.e.*, $w(a(e,c))$ in Eq. 5), respectively.

Instance	Score	Top-4 concepts
airplane	$P(c\|e)$	*vehicle, technology, place, aircraft*
	$P(e\|c)$	*transportation vehicle, bulk-carrying vehicle, mobile source, real time system*
	ours	*vehicle, aircraft, transportation vehicle, mobile source*
coca-cola	$P(c\|e)$	*company, brand, corporation, client*
	$P(e\|c)$	*soft drink, global brand, marketer, corporate giant*
	ours	*corporation, soft drink, company, global brand*

each graph to consider two-way direction between instance and concept, and then connect the variables to each other for modeling full-dependencies. Finally, we propose a new probability estimation model for concepts from this graph.

Constructing MRFs Inspired by (Wang, Wang, and Hu 2014), we first overcome **L2** by weighting an edge between instance $e \in U$ and concept $c \in V$, as the combination of $P(c|e)$ and $P(e|c)$ (*i.e.*, two directions between concept and instance). Specifically, we multiply $P(c|e)$ and $P(e|c)$ to represent a roundtrip random walk probability between e and c. Formally, for an edge $a(e,c)$, the edge weight, $w(a(e,c))$, is computed by multiplying concept and instance typicality:

$$w(a(e,c)) = \begin{cases} 1 & c = e \\ P(c|e) \times P(e|c) & c \neq e \end{cases} \tag{5}$$

$c = e$ represents the case when the example indicates already the most appropriate concept. For example, in Table 1, "company" is both an appropriate concept and also an observed instance for `Employer` FE. In this case, we avoid conceptualizing it to *business* with high $P(c|e) \times P(e|c)$.

Otherwise (*i.e.*, $c \neq e$), with this edge score, we can give low scores to concepts that are either too general or too specific. Table 3 contrasts the examples of the concepts ranked by $P(c|e)$, $P(e|c)$, and $P(c|e) \times P(e|c)$, respectively. In particular, for instance airplane, $P(c|e)$ conceptualizes to *technology* and *place* that are too general, and $P(e|c)$ conceptualizes to *bulk-carrying vehicle* and *real time system* that are too specific. On the other hand, $P(c|e) \times P(e|c)$ leads to more appropriate concepts such as *vehicle*, *aircraft*, *transportation*, and *mobile source* than $P(c|e)$ and $P(e|c)$.

After weighting edges, we add missing edges between concept-instance, instance-instance, and concept-concept pairs for considering full-dependence. More specifically, three edge types are defined as: concept-instance, $a(e,c)$, instance-instance, $a(e_i,e_j)$, and concept-concept, $a(c_i,c_j)$.

The weight of a missing concept-instance edge, $w(a(e,c))$, is computed as the average of instance neighbor similarity and the edge weights between c and the instance neighbors of e (Bliss et al. 2014) as:

$$w(a(e,c)) = \frac{\sum_{e_i \in n^I(e)} (w(a(e,e_j)) \times w(a(e_i,c)))}{|n^I(e)|} \tag{6}$$

where $|n^I(e)|$ is the number of the instance neighbors of e in the same cluster. The weight of a missing instance-instance edge, $w(a(e_i,e_j))$, and a concept-concept edge, $w(a(c_i,c_j))$, are computed as cosine similarity of their concept or instance vectors (Adamic and Adar 2001):

$$w(a(e_i,e_j)) = \frac{\sum_{m=0}^{n^I(e)} w(a(e_i,c_m)) \times w(a(e_j,c_m))}{\sqrt{w(a(e_i,c_m))^2} \times \sqrt{w(a(e_j,c_m))^2}} \tag{7}$$

$$w(a(c_i,c_j)) = \frac{\sum_{m=0}^{n^I(e)} w(a(e_m,c_i)) \times w(a(e_m,c_j))}{\sqrt{w(a(e_m,c_i))^2} \times \sqrt{w(a(e_m,c_j))^2}} \tag{8}$$

For probability estimation, the edge weights are normalized to sum to 1. Thus, the edge weights between two nodes are used as the probability between them (*e.g.*, $P(e,c) = w(a(e,c))$, $P(e_i,e_j) = w(a(e_i,e_j))$, and $P(c_i,c_j) = w(a(c_i,c_j))$).

Posterior Probability Estimation From each graph for the cluster t with instances $e \in U'$ and concepts $c \in V'$, we estimate the posterior probability of concept variable c with considering neighborhoods of the concept. Formally, the posterior probability, $P(c,t)$, is computed as:

$$P(c,t) \propto \prod_{c_i \in V'} \Psi_1(c,c_i) \prod_{e_i,e_j \in U'} \Psi_2(c,e_i,e_j) \tag{9}$$

where $\Psi_1(\cdot)$ and $\Psi_2(\cdot)$ are potential functions.

$\Psi_1(c,c_i)$ is defined as $P(c,c_i)$ on pairs of c and concept neighbors c_i, which considers concept-concept relationships. For example, *job* and *occupations* are synonymous, and observations of an instance to these concepts can be aggregated, to be more robust to missing observations. $\Psi_2(c,e_i,e_j)$ is defined from 3-cliques having two instances and one concept. A 3-clique of multiple instance has been considered as an effective approximation of disambiguated unit cluster (Bordag 2006), which we simulate as a 3-clique of multiple instances sharing the common concept c, *i.e.*, $\Psi_2(c,e_i,e_j)$.

For the posterior probability of c, while Bayes only uses the relationships from concept to instance $P(e|c)$, our model uses the neighborhood relationships between variables.

Approximation of $P(c,t)$ We approximate $P(c,t)$ (Eq. 9) to reduce computations for $\Psi_1(c,c_i)$ and $\Psi_2(c,e_i,e_j)$.

Approximating $\Psi_1(c,c_i)$: Since there are lots of concepts, computing $P(c_i,c_j)$ for every pair is expensive. We have already clustered concept synonyms at Section 4.1. This concept clustering reduces the number of concepts (49.5% concepts are reduced) and obtains the accurate prior probability by merging observations of concept synonyms, *e.g.*, n(police,{*job*, *occupation*, *professional*, *worker*}). After this clustering, synonyms are aggregated (Fig. 3a), after which $\Psi_1(c,c_i)$ can be dropped.

Approximating $\Psi_2(c,e_i,e_j)$: We approximate the potential function $\Psi_2(c,e_i,e_j)$ as:

$$\Psi_2(c,e_i,e_j) = P(e_i,c) \times P(e_j,c) \times P(e_i,e_j) \tag{10}$$

This approximation aggregates the probabilities, $P(e_i,c)$, $P(e_j,c)$, and $P(e_i,e_j)$, of the edge weights of 3-cliques to consider instance-instance and concept-instance pairs.

With this approximation, we finally define the concept score for the whole given instances E by summation of probabilities (Lioma 2008):

$$S(c,E) = \sum_{(e_i,e_j,c) \in t} \Psi_2(c,e_i,e_j) \tag{11}$$

We rank concepts by descending order of this score.

Table 4: The ⟨Frame, FE⟩ pairs used for experiments.

Frame	FE
ACTIVITY_FINISH	Time
APPLY_HEAT	Container
APPLY_HEAT	Heating_instruments
ARREST	Authorities
ATTENDING	Agent
BRINGING	Area
EMPLOYING	Employee
EXPORTING	Importing_area
HUNTING	Hunter
RIDE_VEHICLE	Vehicle

Table 5: The examples of annotator categories.

Category	Frame	FE	Concept
Very typical	ATTENDING	Agent	*professional*
	APPLY_HEAT	Container	*cooking utensil*
Typical	APPLY_HEAT	Container	*kitchen utensil*
	RIDE_VEHICLE	Vehicle	*mobile source*
Related	BRINGING	Area	*geographical feature*
	RIDE_VEHICLE	Vehicle	*machine*
Unrelated	HUNTING	Hunter	*book*
	EMPLOYING	Employee	*movie*

5 Experiments

This section validates our conceptualization and selectional preference approaches using extensive experiments. All experiments were carried out on a machine with a Intel Core i3 CPU processor at 3.07GHz and 4GB of DDR3 memory.

5.1 Conceptualization Evaluation

To validate the proposed conceptualization approach, this section measures whether it can identify appropriate concepts for an instance set of FEs. We report our experimental setup and results.

Setup As conceptualization algorithms, we implement the following approaches: Bayes, CL+Bayes (Sec. 3.2), WordNet-based algorithm (Bryl et al. 2012), and Truss+MRF (Sec. 4).

To run these conceptualization approaches on FrameNet examples, we extract head words from instances in the form of phrases by adopting a standard head word technique (Kawahara et al. 2014). For example, we extract "states" from [The few states]$_{Employer}$.

To evaluate these conceptualization results, we use two human annotators to choose ten ⟨frame, FE⟩ pairs in Table 4 and to verify the top-N concepts ranked by each algorithm, because obtaining all the ground-truth concepts from over 2 million concepts for FEs is non-trivial task. We let human annotators to label the top-30 concepts ranked by each algorithm into four categories: very typical, typical, related, and unrelated, which have the scores, 1, 2/3, 1/3, and 0, respectively. Table 5 shows examples of the concepts categorized by the annotators.

With these human annotated scores as ground-truth, we use the precision and recall of top-N concepts (Lee et al. 2013): $\text{P@}N = \frac{1}{N}\sum_{i=1}^{N} \text{score}_i$ and $\text{R@}N = \frac{\#\text{ very typical concepts in top-}N}{\#\text{very typical concepts in top-50 of the all algorithms}}$ where score_i is the score of the i^{th} concept.

Results Table 6 compares the precision and recall of Truss+MRF with those of baselines. Truss+MRF achieves the highest precision and recall for all N by consistently achieving precision near 0.8. In human annotation, this score corresponds to finding mostly "typical" or "very typical" concepts. For example, for *Vehicle* FE, Truss+MRF extracts the concepts related to vehicle such as *vehicle, mobile source, airborne vehicle*, and *transportation vehicle*.

In contrast, the precision of CL+Bayes is lower than Bayes, which is not consistent with (Song et al. 2011). We find the reason is due to ambiguous instances, such as apple, often forming a singleton cluster. In our problem of FE conceptualization with many terms, such ambiguous instances are more likely to appear. Meanwhile, the recall of WordNet is extremely low, due to sparseness of WordNet concepts (Tonelli et al. 2012), which stops growing for $N > 50$.

In conclusion, Truss+MRF significantly outperforms CL+Bayes in both efficiency and effectiveness: Truss+MRF achieves 608.33 times speed-up ($\frac{4.2\text{ days}}{9.9\text{ minutes}}$) and 0.479 improvement in precision and 0.184 in recall.

However, the quantitative accuracy analysis was limited to 10 frames, as labeling process is labor-intensive, as similarly decided in (Lee et al. 2013) and (Wang et al. 2015) conducted similar evaluations restricted to 12 concepts and six terms, respectively. We thus release our results for all roles for qualitative evaluation of the results[4].

5.2 Pseudo-disambiguation Evaluation

This section evaluates our conceptualization in the end task of computing selectional preference.

Setup As selectional preference algorithms, we use the following two algorithms: topic model-based algorithm (Ritter, Mausam, and Etzioni 2010)[5] and our concept-based algorithm. To evaluate selectional preference, a widely adopted task has been a pseudo-disambiguation task use in (Erk 2007; Ritter, Mausam, and Etzioni 2010) of automatically generating positive ground-truth using frequent co-occurring instances from a corpus then negative ground-truth by pairing with random instances. This evaluation is conducted for two relation-argument pairs $(relation, arg_{sub})$ and $(relation, arg_{obj})$ where arg_{sub} is the subject argument and arg_{obj} is direct object argument of a relation $relation$ (*i.e.*, verb). For example, for a given sentence "*the few states employed chemical weapons*", arg_{sub} is [the few states] and arg_{obj} is [chemical weapons] of a relation *employed*. As the corpus, we use the SemEval 2010 (task 10) dataset (Ruppenhofer et al. 2010) due to its high quality sentences used for a gold standard dataset of semantic role labeling.

We report precision and recall:

[4] The entire results are released at http://karok.postech.ac.kr/FEconceptualization.zip.

[5] This algorithm is downloaded from https://github.com/aritter/LDA-SP

Table 6: The average P@N and R@N over the FEs in Table 4.

Approach	Top-10		Top-20		Top-30		Top-40		Top-50	
	P@10	R@10	P@20	R@20	P@30	R@30	P@40	R@40	P@50	R@50
Bayes	0.640	0.091	0.558	0.146	0.516	0.200	0.492	0.275	0.455	0.317
CL+Bayes	0.393	0.074	0.342	0.104	0.341	0.146	0.307	0.172	0.303	0.210
WordNet	0.687	0.033	0.644	0.045	0.631	0.038	0.631	0.038	0.631	0.038
Truss+MRF	**0.852**	**0.121**	**0.832**	**0.257**	**0.810**	**0.325**	**0.794**	**0.361**	**0.782**	**0.394**

Table 7: Results of pseudo-disambiguation evaluation when recall is 0.2, 0.3, and 0.6.

Approach	Recall = 0.2		Recall = 0.3		Recall = 0.6	
	Prec.	Recall	Prec.	Recall	Prec.	Recall
Topic model	0.66	0.21	0.56	0.29	0.54	0.60
Ours	**0.82**	0.21	**0.74**	0.30	**0.60**	0.61

$\text{Precision} = \frac{\#\text{ positive instances labeled}}{\#\text{ positive instances labeled}+\#\text{ negative instances labeled}}$ and $\text{Recall} = \frac{\#\text{ positive instances labeled}}{\#\text{ total positive instances}}$.

In particular, for each verb, *e.g.*, *employ*, we first identify frames of corresponding frame (*e.g.*, EMPLOYING and USING). For computing plausibility score of $arg_{obj} = weapon$ for each FE, we compare its semantic similarity with FE sets in the same role (using Eq. 11). We identify the frame with a higher score as a potential match, and label positive (if score is over threshold) and negative (otherwise).

Results Table 7 reports results for varying thresholds. Note the accuracy of topic model is significantly lower than published results in (Ritter, Mausam, and Etzioni 2010) due to limited coverage in SemEval task with many named entities, as also observed in Table 2. In contrast, our algorithm achieve up to 0.18 in precision at the same recall.

Acknowledgement

This research was supported by the MSIP (The Ministry of Science, ICT and Future Planning), Korea and Microsoft Research, under ICT/SW Creative research program supervised by the IITP (Institute for Information & Communications Technology Promotion) (IITP-2015-R2215-15-1009). S. Hwang is a corresponding author.

References

Adamic, L., and Adar, E. 2001. Friends and neighbors on the web. *Social Networks*.

Ames, B. P. W. 2012. Guaranteed clustering and biclustering via semidefinite programming. *CoRR*.

Bliss, C. A.; Frank, M. R.; Danforth, C. M.; and Dodds, P. S. 2014. An evolutionary algorithm approach to link prediction in dynamic social networks. In *Journal of Computational Science*.

Bordag, S. 2006. Word sense induction: Triplet-based clustering and automatic evaluation. In *EACL*.

Bryl, V.; Tonelli, S.; Giuliano, C.; and Serafini, L. 2012. A novel framenet-based resource for the semantic web. In *SAC*.

Cohen, J. 2008. Trusses: Cohesive subgraphs for social network analysis. *Technical report, National Security Agency.*

Das, D.; Martins, A. F. T.; and Smith, N. A. 2012. An exact dual decomposition algorithm for shallow semantic parsing with constraints. In *SemEval*.

Erk, K. 2007. A simple, similarity-based model for selectional preferences. In *ACL*.

Fillmore, C. J.; Johnson, C. R.; and Petruck, M. R. 2003. Background to framenet. *International Journal of Lexicography*.

Fillmore, C. J. 1982. Frame semantics. *Linguistics in the Morning Calm*.

Hearst, M. A. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING*.

Kawahara, D.; Peterson, D. W.; Popescu, O.; and Palmer, M. 2014. Inducing example-based semantics frames from a massive amount of verb uses. In *ACL*.

Lee, T.; Wang, Z.; Wang, H.; and won Hwang, S. 2013. Attribute extraction and scoring: A probabilistic approach. In *ICDE*.

Li, P.; Wang, H.; Zhu, K. Q.; Wang, Z.; and Wu, X. 2013. Computing term similarity by large probabilistic isa knowledge. In *CIKM*.

Lidstone, G. J. 1920. Note on the general case of the bayes-laplace formula for inductive or a posteriori probabilities. *Transactions of the Faculty of Actuaries*.

Lioma, C. A. 2008. Part of speech n-grams for information retrieval. In *Ph. D. Thesis, University of Glasgow, Scotland, UK*.

Pizzato, L. A., and Mollá, D. 2008. Indexing on semantic roles for question answering. In *IRQA*.

Resnik, P. 1996. Selectional constraints: An information-theoretic model and its computational realization. *Cognition*.

Ritter, A.; Mausam; and Etzioni, O. 2010. A latent dirichlet allocation method for selectional preferences. In *ACL*.

Ruppenhofer, J.; Sporleder, C.; Morante, R.; Baker, C.; and Palmer, M. 2010. Semeval-2010 task 10: Linking events and their participants in discourse. In *Proceedings of the 5th International Workshop on Semantic Evaluation*.

Song, Y.; Wang, H.; Wang, Z.; Li, H.; and Chen, W. 2011. Short text conceptualization using a probabilistic knowledgebase. In *IJCAI*.

Tonelli, S.; Bryl, V.; Giuliano, C.; and Serafini, L. 2012. Investigating the semantics of frame elements. In *EKAW*.

Wang, Z.; Zhao, K.; Wang, H.; Meng, X.; and Wen, J.-R. 2015. Query understanding through knowledge-based conceptualization. In *IJCAI*.

Wang, Z.; Wang, H.; and Hu, Z. 2014. Head, modifier, and constraint detection in short texts. In *ICDE*.

Wu, W.; Li, H.; Wang, H.; and Zhu, K. Q. 2012. Probase: a probabilistic taxonomy for text understanding. In *SIGMOD*.