Integration of gene normalization stages and co-reference resolution using a Markov logic network

Hong-Jie Dai\textsuperscript{1,2,\ast}, Yen-Ching Chang\textsuperscript{2,3}, Richard Tzong-Han Tsai\textsuperscript{4,\ast} and Wen-Lian Hsu\textsuperscript{1,2,\ast}

\textsuperscript{1}Department of Computer Science, National Tsing-Hua University, Hsinchu, \textsuperscript{2}Intelligent Agent Systems Lab., Institute of Information Science, Academia Sinica, \textsuperscript{3}Department of Life Sciences and Institute of Genome Sciences, National Yang-Ming University, Taipei and \textsuperscript{4}Department of Computer Science and Engineering, Yuan Ze University, Chung-Li, Taiwan, R.O.C.

Associate Editor: Jonathan Wren

\textbf{ABSTRACT}

\textbf{Motivation:} Gene normalization (GN) is the task of normalizing a textual gene mention to a unique gene database ID. Traditional top performing GN systems usually need to consider several constraints to make decisions in the normalization process, including filtering out false positives, or disambiguating an ambiguous gene mention, to improve system performance. However, these constraints are usually executed in separate stages and cannot use each other’s input/output interactively. In this article, we propose a novel approach that employs a Markov logic network (MLN) to model the constraints used in the GN task. Firstly, we show how various constraints can be formulated and combined in an MLN. Secondly, we are the first to apply the two main concepts of co-reference resolution—discourse salience in centering theory and transitivity—to GN models. Furthermore, to make our results more relevant to developers of information extraction applications, we adopt the instance-based precision/recall/F-measure (PRF) in addition to the article-wide PRF to assess system performance.

\textbf{Results:} Experimental results show that our system outperforms baseline and state-of-the-art systems under two evaluation schemes. Through further analysis, we have found several unexplored challenges in the GN task.

\textbf{Contact:} hongjie@iis.sinica.edu.tw

\textbf{Supplementary information:} Supplementary data are available at Bioinformatics online.

Received on April 15, 2011; revised on June 2, 2011; accepted on June 3, 2011

\section{INTRODUCTION}

The task of recognizing named entities in text, i.e. identifying words/phrases that indicate the presence of entities and associating them with their corresponding semantic types has been studied extensively. In biomedical fields, the most commonly used type of named entity recognition is gene mention recognition. In the largest public biomedical text-mining competitions, BioCreative I and II, the top gene mention recognition systems achieve F-scores of \textasciitilde 89\% (Li et al., 2009; Smith et al., 2008).

However, gene mention recognition results are still hard to apply in real-world research applications because of two main issues: name variation and entity ambiguity (Khalid et al., 2008). Name variation is when one gene is referred to by many textual expressions. For example, in Figure 1, the authors refer to the gene ‘EntrezGene ID 966’ as ‘CD95’, ‘membrane inhibitor of reactive lysis (MIRL)’, ‘protectin’, and ‘(MIRL)’. Entity ambiguity is when the same name can refer to more than one gene. For example, a search for the name ‘CD59’ in EntrezGene returns 377 matches belonging to over 10 species.

To spur development in regards to the above two issues, BioCreative has held several open competitions for the task gene normalization (GN), mapping recognized gene mentions to standard database IDs, such as EntrezGene or UniProt IDs (Morgan et al., 2008). Continuing with the example in Figure 1, a GN system must normalize the human gene ‘CD59’ and all its instances in the first sentence to the EntrezGene ID 966. However, ‘CD59’ in the title must be normalized to 12,509 because it is a murine gene. BioCreative has also curated and released standard evaluation datasets as part of its GN tasks, and many novel and useful approaches have come out of the BioCreative workshops.

Generally speaking, after gene mention recognition, the current top-performing systems include three main steps: (i) candidate ID matching, (ii) false positive (FP) filtering and (iii) disambiguation. Some research only focuses on improving one of these steps. For example, Tsuruoka et al. (2007) utilized logistic regression to improve the accuracy of candidate IDs matching. Xu et al. (2007) proposed a knowledge-based disambiguation approach that combines features from text and knowledge sources via an information retrieval method. Crim et al. (2005) used the maximum entropy model to classify valid IDs from candidate ID lists. Hakenberg et al. (2008) and Dai et al. (2010) collected external knowledge for each gene, such as chromosome locations, gene ontology terms, etc. and calculated the likelihoods stating the
similarity of the current text with the knowledge to improve the disambiguation performance. Wang et al. (2010) focused on one source of entity ambiguity, model organism, and developed a corpus for organism disambiguation. For entities that have no corresponding IDs (e.g. ‘C9 complex’ in Fig. 1), Dai et al. (2010) compiled a blacklist from several data sources and dynamically updated the list with full name/abbreviation information found in context to filter out FPs. Hakenberg et al. (2008) employed an isolated stage to filter out FPs, including protein families, groups, or complexes.

Our contributions to GN system development in this article are 3-fold. The first of these contributions is using co-reference information (i.e. whether different mentions refer to the same entity in the same discourse) to boost predictive accuracy. Most previous GN systems do not consider dependencies among gene mentions across sentences in the same article. Here, we propose to model these dependencies in our GN system. Our approach is based on the two main ideas that have been used in co-reference resolution: 

\[ \text{salience in centering theory (Grenz et al., 1995) and transitivity (Ng, 2005).} \]

Discourse salience is a phenomenon that in a given discourse, there is precisely one entity that is the center of attention. Such entity is mentioned over and over again and makes it more salient than others. We can utilize this phenomenon to improve the normalization confidence. Suppose that \( x \) is a candidate ID for several gene mentions in a discourse, we can then assume that \( x \) is more salient than other IDs. If we can normalize one of these mentions, \( m \), to \( x \) with high confidence, then we are more likely to be able to normalize all the other mentions to \( x \) as well. Continuing with the example shown in Figure 1, if ID:966 is a candidate ID for the gene mention ‘CD59’ and all its instances in the first sentence (‘membrane inhibitor of reactive lysis’, ‘MIRL’ and ‘protectin’), we can then assume that ID:966 is more salient than other candidate IDs. We can normalize the mention, ‘MIRL’, to EntrezGene ID 966 with high confidence, because a search for the name in EntrezGene returns only one match. We are, therefore, able to normalize all the other mentions with more confidence, such as the mention ‘CD59’ with 377 ambiguous IDs, to ID:966 as well, because they are in the same discourse and ID:966 is more salient than others.

Similarly, the idea of transitivity allows us to express the concept that if two gene mentions refer to the same gene, and one mention has been normalized to an ID, the other should also be normalized to the same ID. Using the transitivity property, the two ambiguous gene mentions with the same name ‘CD59’ in the second sentence can be normalized. Salience and transitivity have been used to improve the normalization confidence. Because a search for the name in EntrezGene returns only one match. We are, therefore, able to normalize all the other mentions to this same ID with high confidence and use the salience property can result in error propagation. Continuing with the example shown in Figure 1, if ID:966 is a candidate ID for the gene mention ‘CD59’ and all its instances in the first sentence (‘membrane inhibitor of reactive lysis’, ‘MIRL’ and ‘protectin’), we can then assume that ID:966 is more salient than other candidate IDs. We can normalize the mention, ‘MIRL’, to EntrezGene ID 966 with high confidence, because a search for the name in EntrezGene returns only one match. We are, therefore, able to normalize all the other mentions with more confidence, such as the mention ‘CD59’ with 377 ambiguous IDs, to ID:966 as well, because they are in the same discourse and ID:966 is more salient than others.

Our second contribution is integrating FP filtering and disambiguation into a joint inference model. Most previous works employed separate stages to execute FP-filtering and disambiguation. However, a separate-stage approach ignores possible dependencies between FP-filtering and disambiguation and can result in error propagation. Continuing with the example shown in Figure 1, a separate-stage approach is likely to run into trouble. As described above, in the disambiguation stage, we can normalize ‘MIRL’ to ID:966 with high confidence and use the salience property to normalize the others. Unfortunately, a separate FP-filtering stage may filter out the entity mention ‘MIRL’ because it is listed as an abbreviation of an organization name, Mineral Industry Research Laboratory. If filtering is executed first and MIRL is removed, the ID:966 will no longer be considered salient, and normalizing the other mentions will not be so easy. With a joint inference process, we can carry out both FP-filtering and disambiguation tasks simultaneously to avoid this type of error propagation.

Joint models have become popular in natural language processing (NLP) recently, because they allow different NLP tasks to be carried out simultaneously. This makes it possible for features and constraints to be shared among tasks. The Markov logic network (MLN) (Richardson and Domingos, 2006) is a joint model which combines first-order logic and Markov networks. Our MLN model unifies the FP-filtering, co-reference resolution and disambiguation stages, and simultaneously exploits contextual information, co-reference information and filtering constraints.

Our third contribution is to define a new GN evaluation metric for information extraction (IE) applications. Existing GN evaluation schemes mainly aim to assess system performance in terms of effectiveness for database curation (Morgan et al., 2008). For each article, they usually compare a list of gene IDs output by the system to a gold standard list for that article. We refer to this evaluation scheme as article-wide resolution. For IE applications, such as the biomolecular event extraction task in the BioNLP shared task, however, GN needs to be much more accurate. For each instance of an ambiguous gene mention, the correct ID must be determined for the dependent application to make the correct inferences. Therefore, to make our results more relevant to the developers of IE applications, we assess our system at a finer-grained resolution, instance by instance in addition to article-wide resolution.

2 METHODS

2.1 Markov logic

In first-order logic, formulae consist of four types of symbols: constants, variables, functions and predicates. Constants represent objects in a specific domain (e.g. gene mention: CD59, MIRL, etc. or Entrez IDs). Variables (e.g. \( x, y \)) range over the objects. Predicates represent relationships among objects (e.g. \( \text{interact} \) or attributes of objects (e.g. \( \text{isLocalizedTo} \)). Constants and variables may belong to specific types. An atom is a predicate symbol applied to a list of arguments, which may be constants or variables (e.g. \( \text{isLocalizedTo} \left( \text{CD59}, x \right) \) or \( \text{interact} \left( \text{CD59}, y \right) \)). A ground atom is an atom whose arguments are all constants (e.g. \( \text{isLocalizedTo} \left( \text{CD59}, 966 \right) \)). A world is an assignment of truth values to all possible ground atoms. A knowledge base (KB) is a partial specification of a world; each atom in it is true, false or unknown.

A Markov network represents the joint distribution of a set of variables \( X = \left(X_1, X_2, \ldots, X_n \right) \subset \mathcal{X} \) as a product of factors: \( P(X=x)=\prod_k \Phi_k(x_k) \), where each factor \( \Phi_k \) is a non-negative function of a subset of the variables \( x_k \) and \( Z \) is a normalization constant. As long as \( P(X=x)>0 \) for all \( x \), the distribution can be equivalently represented as a log-linear model: \( P(X=x)=\exp\left(\sum_i \lambda_i w_i(x_i)\right) \), where the features \( w_i(x) \) are arbitrary functions of (a subset of) the variables' states.

An MLN is a set of weighted first-order formulae. Together with a set of constants representing objects in the domain, it defines a Markov network with one variable per ground atom and one feature per ground formula. The probability distribution over possible worlds \( x \) is given by \( P(X=x)=\frac{1}{Z} \exp\left(\sum_i \lambda_i w_i(x_i)\right) \), where \( Z \) is the partition function. \( F \) is the set of all first-order formulae in the MLN, \( g_i(x) \) is the set of groundings of the \( i \)-th first-order formula and \( g_i(x)=1 \) if the \( i \)-th ground formula is true and \( g_i(x)=0 \) otherwise. Markov logic enables us to compactly represent complex models in non-i.i.d. domains. General algorithms for learning and inference in Markov logic are discussed in Richardson and...
As shown in Table 1, there are numerous observed predicates defined for the disambiguation process. These predicates capture several types of information that can be divided into two groups: (i) gene profile information: information extracted from manually curated knowledge resources that are relevant to the gene ID; and (ii) non-profile information: information directly derived from the gene’s context in the given abstract.

<table>
<thead>
<tr>
<th>Table 1. Main observed predicates for GN</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Profile based</strong></td>
</tr>
<tr>
<td>isCandidateOf(id, i; id) is a candidate ID of the i-th gene mention</td>
</tr>
<tr>
<td>hasChromosomeInfo((i, id, sd); id) the chromosome location information of the i-th gene mention, which has the candidate identifier id, can be found in the surrounding text in the sentence distance sd.</td>
</tr>
<tr>
<td>hasPPIPartnerRank(i, id, r); r) the i-th gene mention has an identifier candidate id that interacts with the rank-r numbers of unambiguous IDs found in the current content among all r’s candidates.</td>
</tr>
<tr>
<td>hasPPIKeyword(w) the word w is a protein-protein interaction keyword.</td>
</tr>
</tbody>
</table>

Please refer to the Supplemental Material for the full list of predicates.

2.4.1 Gene profile-based disambiguation We define four predicates which have been used in previous researches (Dai et al., 2010; Hakenberg et al., 2008; Lai et al., 2009) to capture recognized genes’ profile information, including chromosome location, protein-protein interactions (PPIs), tissue type and gene ontology. For example, the predicate hasChromosomeInfo((i, id, sd) indicates that the chromosome location information of the i-th gene mention, which has the identifier id as its candidate ID, can be found in the surrounding text at the range sd. Applying this predicate to the sentence: ‘The human UBIQLN3 gene was mapped to the 11p15 region of chromosome 11.’

The mention UBIQLN3 must be normalized to the EntrezGene ID:50613 because 50613’s chromosome location, 11p15, is found in the same sentence. The formula describing the relation of hasChromosomeInfo and isNormalizedTo is defined as follows:

\[ \text{hasChromosomeInfo}(i, id, sd) \Rightarrow \text{isNormalizedTo}(i, id) \]

Here, we can see that there is an additional parameter, sd, in the predicate hasChromosomeInfo. id, indicating where the chromosome information corresponding to id locates, has two possible values: 0 indicates the id’s chromosome information is located in the same sentence as i. Otherwise, id is 1. The ‘+’ notation in the above formula indicates that the MLN must learn a separate weight for each grounded variable (id). For example, hasChromosomeInfo((i, id, 0) and hasChromosomeInfo((i, id, 1) are given two different weights in our MLN model after training. Based on the chromosome information recorded in EntrezGene, we use regular expression patterns, such as ‘(\d+)(\d+)(\d+)[pq](\d+)’, ‘(\d+)(\d+)(\d+)’ to determine whether the chromosome information for a given gene mention exists in the context.

The PPI information can be used in disambiguating a gene mention as follows. Based on the PPI recorded in the database, we assume that the id, which interacted with the most unambiguous IDs, is the most likely id that can be normalized. We define the predicate hasPPIPartnerRank and isNormalizedTo is:

\[ \text{hasPPIPartnerRank}(i, id, r) \Rightarrow \text{isNormalizedTo}(i, id) \]

as follows. Based on the PPI recorded in the database, we assume that the id, which interacted with the most unambiguous IDs, is the most likely id that can be normalized. We define the predicate hasPPIPartnerRank and isNormalizedTo is:

\[ \text{hasPPIPartnerRank}(i, id, r) \Rightarrow \text{isNormalizedTo}(i, id) \]

as follows. Based on the PPI recorded in the database, we assume that the id, which interacted with the most unambiguous IDs, is the most likely id that can be normalized. We define the predicate hasPPIPartnerRank and isNormalizedTo is:

\[ \text{hasPPIPartnerRank}(i, id, r) \Rightarrow \text{isNormalizedTo}(i, id) \]
One can see that there are two predicates in this formula that check if the article contains protein–protein interaction (PPI) keywords. Two similar predicates, hasGOTermRank and hasTissueTermRank, represent the concept that i should be normalized to the id with the largest number of corresponding gene ontology term or tissue term found in the context. For estimating the tissue and the gene ontology term counts, we collected terms from the Human Protein Reference Database and the gene ontology database, using the exact matching approach to calculate the matching counts in the context.

For PPI, we further define the following formula to capture the dependency that a gene mention j should be normalized to id if another gene mention i has been normalized to id and i-1 forms an interaction with i:

\[
\text{Formula 4:} \quad \text{hasPPIkeyword}(w, w') \land \text{isCandidateOf}(id, j) \land \text{isNormalizedTo}(i, id) \Rightarrow \text{isNormalizedTo}(j, id)
\]

2.4.2 Non-profile-based disambiguation

If the context does not contain gene profile-related information, non-profile information can be used. For example, a gene mention j may sometimes be followed by its variant i (abbreviation or full name). Usually, the variant i is put in parentheses. If i can be uniquely mapped to id, it is very likely that j is also normalized to id. The actual formula is shown as follows:

\[
\text{Salience formula:} \quad i < j \land \text{hasUnigramBetween}(i, j, w) \land w = \text{"i"} \land \text{hasCandidateOf}(id, i) \Rightarrow \text{isNormalizedTo}(j, id)
\]

Finally, the salience property described in the introduction section can be exploited in disambiguation as follows:

\[
\text{Salience formula:} \quad i < j \land \text{isNormalizedTo}(id, i) \land \text{isCandidateOf}(id, j) \Rightarrow \text{isNormalizedTo}(i, id)
\]

In other words, if the identifier id is normalized to a gene mention i that precedes the current mention j, and id is a candidate of j, then the current mention j should also be normalized to id.

2.5 Formulae for FP-filtering

Ideally, we should be able to treat all recognized gene mentions and their IDs as candidates, and proceed directly to the disambiguation task. However, it is not always the case, because the employed recognizer may generate FP gene mentions. To capture the concept in our model, we define the predicate shouldBeNormalized, which indicates that the i-th gene mention of the article should be normalized to an ID. We then employ the following formula to ensure that, whenever i is normalized to an identifier id, a gene mention should be normalized.

\[
\text{Formula 5:} \quad \text{hasCandidateOf}(id, i) \Rightarrow \text{isNormalizedTo}(i, id)
\]

2.6 Formulae for co-reference

In addition to GN, our MLN model infers whether or not the gene mentions i and j are the same instances. The predicate coReference(i, j) is used to indicate that the two mentions i and j are the same instances. By jointly predicting co-references and GN in our model, we can define the following formula

\[
\text{coReference(i, j)} \Rightarrow \text{isNormalizedTo}(i, id) \land \text{isNormalizedTo}(j, id)
\]

One of the obvious advantages of the MLN model is that it can easily be extended to model co-references. Co-references are defined as follows:

\[
\text{coReference}(i, j) \Rightarrow \text{isNormalizedTo}(i, id) \land \text{isNormalizedTo}(j, id)
\]

2.2 Main observed predicates for FP-filtering

\[
\begin{align*}
\text{hasGeneName}(i, w) & : \text{the name of the i-th gene mention is w} \\
\text{hasFirstWord}(i, w) & : \text{the first word of the i-th gene mention is w} \\
\text{SpeciesTerm}(w) & : \text{the word w is a species term.} \\
\text{Blacklisted}(n) & : \text{the word sequence n is blacklisted.} \\
\text{containsMoreSpecificMentions}(i) & : \text{the i-th gene mention collocates with more specific gene mentions in the current context.} \\
\text{AllUpperCases}(w) & : \text{the word sequence w are all uppercase.}
\end{align*}
\]

The other formulae are constructed by using the observed predicates defined in Tables 1 and 2 to determine whether i is a true gene mention or not by checking i’s context. For example:

\[
\text{Formula 6:} \quad \text{hasCandidateOf}(id, i) \Rightarrow \text{isNormalizedTo}(i, id)
\]

2.3 Main observed predicates for GN

\[
\begin{align*}
\text{hasGeneName}(i, w) & : \text{the name of the i-th gene mention is w} \\
\text{hasFirstWord}(i, w) & : \text{the first word of the i-th gene mention is w} \\
\text{SpeciesTerm}(w) & : \text{the word w is a species term.} \\
\text{Blacklisted}(n) & : \text{the word sequence n is blacklisted.} \\
\text{containsMoreSpecificMentions}(i) & : \text{the i-th gene mention collocates with more specific gene mentions in the current context.} \\
\text{AllUpperCases}(w) & : \text{the word sequence w are all uppercase.}
\end{align*}
\]

The other formulae are constructed by using the observed predicates defined in Tables 1 and 2 to determine whether i is a true gene mention or not by checking i’s context. For example:

\[
\text{Formula 7:} \quad \text{hasCandidateOf}(id, i) \Rightarrow \text{isNormalizedTo}(i, id)
\]
Table 3. Observed predicates and formulae for co-reference resolution

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>isAliasOf(i, j)</td>
<td>the i-th gene mention is listed as an alias of the j-th gene mention.</td>
</tr>
<tr>
<td>Distance(id, j, dis)</td>
<td>the distance between the i-th and the j-th gene mention is dis.</td>
</tr>
<tr>
<td>InAppositionTo(i, j)</td>
<td>the i-th gene mention is in apposition to the j-th gene mention.</td>
</tr>
</tbody>
</table>

String match feature:

- hasGeneName(id, n) \(\rightarrow\) hasGeneName(id, n) \(\Rightarrow\) isCoreference(i, j)
- Alias feature:
  - isAliasOf(i, j) \(\rightarrow\) hasGeneName(id, n) \(\Rightarrow\) isCoreference(i, j)
- Distance feature: Distance(id, j, dis) \(\Rightarrow\) isCoreference(i, j)
- Apposition feature: InAppositionTo(i, j) \(\Rightarrow\) isCoreference(i, j)

Transitivity formulae: isCoreference(i, j) \(\Rightarrow\) isNormalizedTo(id, id) \(\rightarrow\) ¬∃\(\exists\) id, isNormalizedTo(j, id), id) \(\Rightarrow\) isNormalizedTo(id, id)

Symmetry: ∀i, j isCoreference(i, j) \(\Rightarrow\) isCoreference(j, i)

Reflexivity: ∀i, j isCoreference(i, j) \(\Rightarrow\) isCoreference(i, i)

Transitivity: ∀i, j, k isCoreference(i, j) \(\Rightarrow\) isCoreference(j, k) \(\Rightarrow\) isCoreference(i, k)

3 RESULTS

3.1 Experimental setup

3.1.1 Evaluation schemes We use the standard recall, precision and F-measure metrics (RPP) to evaluate our approach and compare it with other GN methods at two resolutions (article and instance). Article-wide evaluation is based on the standard used in the BioCreative II GN challenge (Morgan et al., 2008), which was designed to determine a GN system’s performance in aiding curation of biological databases. For a given article, the GN system outputs a list of IDs, which is then compared to the gold standard list for the article. The RPP scores are calculated based on the sums of true/false positives/negatives (TP, TN, FP, FN).

Instance-based evaluation measures the GN performance at a finer-grained IL resolution. In contrast to the strict metric, the RPP scores are calculated based on the sums of TP, TN, FP and FN for all instances in the test dataset. We further consider whether the boundaries match those of the normalized identifier’s mention. Under this criterion, an FP could normalize a true gene mention to the wrong ID or a false gene mention to any ID while an FN could normalize a true gene mention to the wrong ID or fail to recognize a true gene mention. In cases where a true gene mention is normalized to the wrong ID, both the FN and FP are increased by 1. For TP/FN/FN, we need to determine when the predicted boundaries match those of the gold standard. Most entity recognition tasks use ‘exact-matching’ as the primary criterion. Under this criterion, a candidate gene mention can only be counted as a TP if both its left and right boundaries fully coincide with the gold answer. However, in a real scenario, a gene mention can be tagged in several ways (e.g. ‘no correlation between serum entity-LH entity’ and ‘no correlation between entity entity-LH entity entity’ are both correct), which are intrinsic to the annotation of any gene mention corpus whether developed by humans or machines, and may depend on the annotator’s perspective. Furthermore, for the GN task, the correctness of the normalized ID is more important than its boundaries. Therefore, we use approximate-matching (Subramaniam et al., 2003) to determine the boundary criterion. For example, a TP is counted when a machine-normalized gene mention is a substring of the gold standard-normalized gene mention or vice versa, and the normalized ID is equal to the gold ID.

3.1.2 Dataset We use the training and test sets (281 and 262 abstracts respectively) released by the BioCreative II GN task. The corpus contains annotations for human genes that are normalized to IDs in EntrezGene database. We chose this dataset rather than the more recent one released as part of the BioCreative III GN task (Lai et al., 2011) because each abstract in the BioCreative II training/test is accompanied by a list of gene IDs and corresponding name strings found in that abstract. The BioCreative III GN dataset, on the other hand, does not include name-string information in these lists, only gene IDs. This lack of name-string information makes it very difficult for our biologists to manually compile a corpus for instance-based evaluation because one ID can correspond to many name strings. Although the gold BioCreative II standard contains each ID’s name string, it does not give the exact location of the corresponding gene mention in the abstract. To obtain instance-based evaluation results, our in-lab biologists annotated the exact locations and the boundaries of the IDs’ gene mentions with automated assistance. The automatic annotation process uses the ID’s name string from the gold standard to tag the entire corpus. Human annotators then corrected the boundaries and normalized results based on the context.

To compile the GN training corpus for our MLN models, we employed a publicly available state-of-the-art GN system released by Lai et al. (2009) to recognize all gene mentions and generate candidate IDs for each entity. Lai’s gene mention recognition system achieved an F-score of 85.8% on the BioCreative II gene mention tagging corpus. For each mention m in a sentence s recognized by Lai’s system and the set of EntrezGene ID candidates for m output by Lai’s system, we searched s for the first human annotated mention n overlapping with m and set n’s human annotated ID as m’s true EntrezGene ID. Other candidates were set as m’s incorrect IDs.

(Please refer to our Supplementary Material for the details of the dataset construction. The compiled corpus would be available at https://sites.google.com/site/hongjiedai/projects.)
We chose Lai’s system, which is the core component of the rank 1 system in BioCreative II 5 interactor normalization task (Dai et al., 2010), because it is the only publicly available state-of-the-art system developed for the BioCreative II GN task. The performance of another open available library, Moara (Neves et al., 2010), is far from the state-of-the-art on the same dataset.

For the FP-filtering corpus, again, for each mention \( m \) in a sentence \( s \) recognized by Lai’s system, we checked whether or not the boundaries of the mention \( m \) matched with the human-annotated boundaries. All matched mentions are regarded as TPs while the others are TN instances. Finally, to generate our co-reference resolution corpus, we simply treated gene mentions generated by Lai’s system containing the corresponding same gold normalized ID as co-references.

3.1.3 System setting In order to compare our MLN-based GN system to separate-stage-based systems, we constructed two stage-based GN systems (Systems 1 and 2). Both stage-based systems as well as our system employed Lai’s system for gene mention recognition and ID matching for each gene mention. Systems 1 and 2 also share two components: FP-filtering and co-reference resolution.

These two components are based on the maximum entropy (ME) model, and are referred to as ME Model1 and Model2. In ME modeling, we formulated both the FP-filtering and the co-reference resolution tasks as classification tasks. Model1 used the features equivalent to the formulae described in Section 2.5 (FP-filtering). Model2 uses the feature functions equivalent to the co-reference resolution formulae described in Section 2.6. Please refer to our Supplementary Material for details.

In the disambiguation stage, Lai’s system (2009) was used in System 1 while the ME-based approach with equivalent features described in Section 2.4 was used in System 2. In System 2, we followed Crim et al.’s approach (2005) to formulate the GN task as a classification problem and transform all formulae described in Section 2.4 except Formula 3 and the Salience formula to binary feature functions. Formula 3 and Salience formula are excluded because they cannot model in ME. We will discuss this in Section 4.1.

For Systems 1 and 2, we employed an additional step to select the optimal set of features with greedy backward sequential selection algorithm (Aha and Bankert, 1995). For each system, the algorithm starts from all features transformed from FP-filtering formulae and repeatedly removes a feature whose removal yields the maximal performance improvement in the overall GN task. The same algorithm is then used to select the optimal set of co-reference features. Note that the feature selection procedure is designed for optimizing the performance of GN not FP-filtering or co-reference resolution. We will discuss this in Section 4.4.

In contrast to the MLN-based GN system, which performs joint inference for FP-filtering, disambiguation and co-reference resolution at once, to carry out the above stages in System 1 and 2, we followed this procedure: after one or several ID matches were found for a gene mention, Model1 was employed for both systems to decide whether to keep the mention or discard it. If the mention was kept, each system’s disambiguation method was then used to select the most appropriate ID for it. In addition, Model2 was employed to recognize the co-references in an article. The co-reference information was fed into the assignment algorithm, which was implemented as a post-processing step in both systems.

To approximate Transitivity Formula described in Section 2.6, the assignment algorithm was implemented as follows. For each non-normalized gene mention \( i \), and its co-reference chain determined by Model1, the algorithm chose the ID with the highest confidence from the co-reference chain, and then assigned it to \( i \). In the next subsection, we discuss the instance-based fine-grained IE results. Then, we derive BioCreative’s evaluation results by simply merging the normalized IDs in all locations and removing duplicated IDs.

3.2 Experiment results Table 4 shows the instance-based results derived on the test set. The first row (no disambiguation/MLN-based: 2.3) assesses the performance without applying any disambiguation approaches of Lai’s system. This result shows the baseline for Systems 1 and 2 for which all mentions with only one candidate ID were directly treated as answers, and entities with more than one candidate ID were discarded. Our MLN-based model achieves exactly the same performance (MLN-based: 2.3) by applied formulae described in Section 2.3 (GN Constraints Formulae), indicating the MLN-based system can simulate the decision. For each configuration, the last column of its corresponding row shows its \( F \)-score improvement over the baseline after employing different GN disambiguation methods, and stage processing or joint inference.

Rows of (a)–(c) compare our MLN-based disambiguation approach, which uses all formulae defined in Sections 2.3 and 2.4 (MLN-based/2.3+2.4), with Lai et al.’s approach (System 1) and the ME-based disambiguation approach (System 2). The (e) and (f) employ the ME Model1 to further filter out FPs. Our MLN-based GN system (d) achieves the same goal by adding on (a) with formulae defined in Section 2.5 that captures the filtering concept.

Rows of (g)–(i) shows the results of the three systems that further exploits the co-reference information. In contrast to the cost of developing another algorithm to combine the co-reference information into the original GN systems (e) and (f), we can achieve the same goal in MLN by simply adding all formulae in Section 2.6 into our joint model (d). Finally, the results derived on the test set using article-wide criterion are shown in Table 5.

From (a)–(c), we can see that with equivalent disambiguation feature setting, MLN outperforms the other two models. Adding
We list their effects in Table 6. This shows that (i) a scientific article often contains information that
results derived on the test set using the article-wide evaluation
in an improved
disambiguation formulae, the recall rate is improved and results
centering theory (see Salience Formula) and transitivity property of
4.1 The effects of discourse salience and transitivity
methods under both two evaluation criteria.
by apparent large margin (3.7 and 10.3%). By
'GOLD' in configuration (e) refers to using the gold co-reference annotations.
configuration (e) + Salience 82.5 67.5 74.2 +7.9
configuration (d) + Salience 79.1 59.4 67.8 +1.5
configuration (a) + Transitivity + GOLD 80.6 67.5 73.5 +7.2
configuration (e) + Transitivity + Sec.2.6 80.5 57.0 66.8 +0.5
without any disambiguation formulae. The recall rate is improved and results
in an improved $F$-score under instance-based criterion. With gold
correlation (see [c] and [f]), the MLN-based system
performance obviously achieves more significant improvements.
This shows that (i) a scientific article often contains information that
appears repeatedly throughout the paper, such as key (salient) genes,
which can be captured by our model; (ii) the transitivity property
in response to tissue injury"
Lai’s gene mention recognition system and the three publically
available systems (http://pages.cs.wisc.edu/~bsettles/abner
http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger and http://
School of medicine/). All separate the first gene mention


\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|}
\hline
\textbf{Configuration} & \textbf{P} & \textbf{R} & \textbf{F} \\
\hline
No disambiguation/MLN-based: 2.3 & 77.3 & 71.4 & 74.2 \\
\hline
(a) MLN-based/2.3+2.4 & 86.1 & 83.0 & 84.5 +10.3 \\
(b) System 1 (Lai et al., 2009) & 82.6 & 83.4 & 83.0 +6.8 \\
(c) System 2 & 88.9 & 79.0 & 83.7 +9.4 \\
(d) MLN-based/ (a)+2.5 & 89.7 & 81.9 & 85.6 +11.33 \\
(e) System 1 + Model$_1$ & 84.7 & 83.4 & 84.0 +9.7 \\
(f) System 2 + Model$_1$ & 96.7 & 74.4 & 84.1 +9.9 \\
(g) MLN-based/(d)+2.6 & 89.9 & 81.6 & 85.5 +11.30 \\
(h) System 1 + Model$_1$ + Model$_2$ & 83.8 & 83.5 & 83.6 +9.4 \\
(i) System 2 + Model$_1$ + Model$_2$ & 96.7 & 74.3 & 84.0 +9.8 \\
\hline
\end{tabular}
\caption{Results derived on the test set using the article-wide evaluation}
\end{table}

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|}
\hline
\textbf{Configuration} & \textbf{P} & \textbf{R} & \textbf{F} \\
\hline
(a) No disambiguation/MLN-based: 2.3 & 80.7 & 56.3 & 66.3 \\
(b) (a) + Salience & 79.5 & 59.0 & 67.7 +1.4 \\
(c) (a) + Transitivity + Sec.2.6 & 80.5 & 57.0 & 66.8 +0.5 \\
(d) (c) + Salience & 79.1 & 59.4 & 67.8 +1.5 \\
(e) (c) + Transitivity + GOLD & 80.6 & 67.5 & 73.5 +7.2 \\
(f) (e) + Salience & 82.5 & 67.5 & 74.2 +7.9 \\
\hline
\end{tabular}
\caption{The effects of salience and transitivity on the test set}
\end{table}

\section{DISCUSSION}

\section{4.1 The effects of discourse salience and transitivity}
In this work, we propose to adopt the salient discourse property of the
centering theory (see Salience Formula) and transitivity property of
correlation relations (see Transitivity Formula) in our MLN model.
We list their effects in Table 6.
As shown in Table 6, by adding these two properties without
any disambiguation formulae, the recall rate is improved and results
in an improved $F$-score under instance-based criterion. With gold
correlation information (see [e] and [f]), the MLN-based system
performance obviously achieves more significant improvements.
This shows that (i) a scientific article often contains information that
appears repeatedly throughout the paper, such as key (salient) genes,
which can be captured by our model; (ii) the transitivity property
in response to tissue injury"
Lai’s gene mention recognition system and the three publically
available systems (http://pages.cs.wisc.edu/~bsettles/abner
http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger and http://
School of medicine/). All separate the first gene mention
We also observe that adding these formulae (Salience and Transitivity) do not have significant effects on article-wide evaluation. According to our analysis, these properties improve the recall in the instance-based evaluation. In contrast, for article-wide, they tend to improve the overall precision; however, it might slightly reduce the recall. This phenomenon is reasonable. For example, after adding Salience formula, gene mentions tend to be normalized to 'salient' IDs. In instance-based evaluation, the salient IDs have higher frequency; therefore, the improvement of normalizing salient IDs can cover the losses caused by disregarding the unsalient IDs. However, in the article-wide evaluation, all entries in an article are counted equally; therefore, the improvement of instance-based evaluation does not transfer to article-wide evaluation.
Finally, in addition to the low $F$-score (66.3\%) of our joint model in co-reference resolution, we observed that ambiguous species descriptions led to a performance gap between (d) and (f). As shown in Formula 5, we detect the corresponding species by checking whether any surrounding word w of a gene mention is a species term [SpeciesTerm(w)]. Adding all formulae related to the [SpeciesTerm(w)] predicate to (d), we found that the precision and $F$-scores for instance-based GN increase by 0.7 and 0.2\%, respectively. This result shows that species ambiguity is an issue when employing discourse salience in GN and adding species-recognition capability to the system can help make up this gap.

\subsection{4.2 Normalizing one mention to multiple IDs}
Another advantage of our model is that it is flexible. The GN task is
usually defined as normalizing a mention to a unique ID. However,
we have observed that there are mentions that cannot be uniquely
normalized. Take the following sentence for example: 'ABC89 protein
appears to be mostly expressed in the Sertoli cells of the seminiferous tubules in mouse and rat testes.'
Baumgartner et al. (2007) also found that gene mentions were usually hidden in some form, such as 'Arf2/3', 'IL-7 and IL15 receptors' and 'SMADs 1, 5 and 8', in an article. The issue is usually solved by employing an additional pre-processing to expand the collapsed ranges (Dai et al., 2010). In our model, if one wishes to normalize a gene mention to more than one ID, one can simply modify the constraint in Formula 2 to increase cardinality, or introducing additional formulae to determine the cardinal constraint dynamically.

\subsection{4.3 Boundary issue in instance-based GN}
Our experiment results raise an interesting question: what causes absolute score differences between fine-grained IE (instance-based) and database curation (article-wide)? Several works have studied the boundary issue in entity recognition (Finkel et al., 2005; Tsai et al., 2006). We have observed that the issue also has a significant effect on the performance of GN. For example, consider the following sentence in the training set (PMID: 9346890):

\textit{Sentence S.1:} '<entityid=3083>-Hepatocyte growth factor (HGF) activator</entity>' is a serine protease responsible for proteolytic activation of '<entityid=3082>-HGF</entity> in response to tissue injury'
Lai’s gene mention recognition system and the three publically
available systems (http://pages.cs.wisc.edu/~bsettles/abner
http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger and http://
School of medicine/). All separate the first gene mention

(id = 3083) into at least one mention, (‘hepatocyte growth factor’ or ‘HGF’). This incorrect boundary leads to errors in GN, and could result in the extraction of an incorrect self-activation event:

An experiment conducted on the test set shows that our MLN model can achieve an F-score of 81.7% in fine-grained IE if we replaced the predicted mentions’ boundaries with their corresponding overlapping gold boundaries. These results show that a hybrid approach may be useful for generating gene mentions for GN. For example, continuing with the Sentence 5.1, if we put it through a syntactic parser like Enju, we find that the adjacent words ‘Hepatocyte growth factor (HGF) activator’ belong to the same noun phrase, which indicates that we can expand the boundary. We plan to address this issue in future work.

4.4 Joint model versus separate-stage models

Compared with the two separate-stage Systems 1 and 2, our MLN-based approach has the following two advantages: (i) it performs several predictions using one model and (ii) it finds the global optimal solution. The first advantage has been illustrated by Meza-Ruiz et al. (2008), which is contrasted with separate-stage systems where several components need to be trained and integrated by different strategies. The second advantage is based on our observation on the training set, employing all features transformed from FP-filtering formulae in the ME Model1 that might be able to achieve the best FP-filtering performance, but it does not guarantee that the final integrated GN performance can also be the best. This is why we need to employ the backward feature selection algorithm to optimize GN performance for the separate-stage systems. The same phenomenon is also appeared when combining the ME Model2 with ME Model1 and different separate-stage disambiguation approaches.

We also observed that each individual component’s F-score in the joint model is higher than that of the separate-stage models. For example, the FP-filtering F-score in MLN joint model (79.5%) is 2.7% higher than the F-scores achieved by separate-stage models. In co-reference resolution, the joint model also achieves a better F-score (66.3% versus 64.9%). These results also state the advantage of joint inference.

5 CONCLUSIONS

In this article, we present a novel approach that employs MLN to model the constraints and decisions in the GN task. Our formulae describe several properties, including gene profile and non-profile-based information, which can be used for GN disambiguation. We use dependencies among IDs to model the discourse salience and the transitivity properties. Moreover, we integrate the FP-filtering and disambiguation steps into a simultaneous process and demonstrate the benefit of predicting gene mentions and their corresponding IDs simultaneously in contrast to the stage-based approaches, which identify mentions first and then normalize them to IDs. We also show the performance boost of exploiting co-reference information in GN. For system evaluation, we propose a new fine-grained scheme that assesses results instance by instance, instead of article by article. Our experiments provide the first gene mention evaluation results from a fine-grained IE perspective and highlight problems that need to be addressed in GN systems, e.g. the assignment of non-unique IDs and the boundary issue.

ACKNOWLEDGMENTS

Authors’ contributions: H.J.D.: developed the MLN-based GN system, designed all the experiments and wrote most of this paper. H.J.D., Y.C.C. and R.T.H.T. discussed and refined the paper. Y.C.C., one of the biologists in our laboratory, annotated the GN corpus for instance-based evaluation. R.T.H.T. and W.L.H. guided the whole project.

Funding: National Science Council under grant NSC99-3112-B-001-005 and NSC98-2221-E-155-060-MY3, the Academia Sinica Investigator Award 95-02 and the research center for Humanities and Social Sciences under grant HS-50-23.

Conflict of Interest: none declared.
H.-J. Dai et al.