BIOINFORMATICS ORIGINAL PAPER
Vol. 27 no. 18 2011, pages 2586–2594
doi:10.1093/bioinformatics/btr358

Data and text mining
Advance Access publication June 17, 2011

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

Hong-Jie Dai[1,2,*], Yen-Ching Chang[2,3], Richard Tzong-Han Tsai[4,*] and Wen-Lian Hsu[1,2,*]

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

Associate Editor: Jonathan Wren

ABSTRACT

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 several 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.

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.

Contact: hongjie@iis.sinica.edu.tw

Supplementary information: Supplementary data are available at *Bioinformatics* online.

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

1 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 ~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', 'protectin', '**m**embrane **i**nhibitor of **r**eactive **l**ysis,' 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.

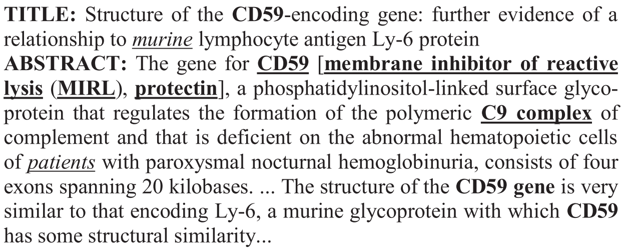
TITLE: Structure of the **CD59**-encoding gene: further evidence of a relationship to *murine* lymphocyte antigen Ly-6 protein
ABSTRACT: The gene for **CD59** [**membrane inhibitor of reactive lysis** (**MIRL**), **protectin**], a phosphatidylinositol-linked surface glycoprotein that regulates the formation of the polymeric **C9 complex** of complement and that is deficient on the abnormal hematopoietic cells of *patients* with paroxysmal nocturnal hemoglobinuria, consists of four exons spanning 20 kilobases. ... The structure of the **CD59 gene** is very similar to that encoding Ly-6, a murine glycoprotein with which **CD59** has some structural similarity...

Fig. 1. A PubMed abstract (PMID: 1381503) discusses the relationship of the gene 'CD59' to other lymphocyte antigens.

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

[*]To whom correspondence should be addressed.

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: *salience* in centering theory (Grosz *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 performance of biomedical relation extraction by Yoshikawa *et al.* (2010), but have not been studied in GN. We will show how to employ them in GN and evaluate their effects.

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 EntrezGene IDs). *Variables* (e.g. x, y) range over the objects. *Predicates* represent relationships among objects (e.g. *interact_with*) or attributes of objects (e.g. *isNormalizedTo*). 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. *isNormalizedTo*('CD59′, x) or *interact_with*('CD59′, y)). A *ground atom* is an atom whose arguments are all constants (e.g. *isNormalizedTo*('CD59′, 966)). 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 $X=(X_1,\ldots,X_n)\in\mathcal{X}$ as a product of factors: $P(X=x)=\frac{1}{Z}\prod_k f_k(x_k)$, where each factor f_k is a non-negative function of a subset of the variables x_k and Z is a normalization constant. As long as $\mathrm{P}(X=x)>0$ for all x, the distribution can be equivalently represented as a log-linear model: $\mathrm{P}(X=x)=\frac{1}{\mathrm{Z}}\exp(\sum_i w_i g_i(\mathsf{x}))$, where the features $g_i(x)$ 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 $\mathrm{P}(X=x)=\frac{1}{\mathrm{Z}}\exp(\sum_{i\in F}\sum_{j\in G_i} w_i g_j(x))$ where Z is the partition function, F is the set of all first-order formulae in the MLN, g_j is the set of groundings of the i-th first-order formula and $g_j(x)=1$ if the j-th ground formula is true and $g_j(x)=0$ 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

Domingos (2006). We employ the beast toolkit (http://code.google.com/p/thebeast/) to implement our MLN model. It uses 1-best Margin Infused Relax online learning Algorithm (MIRA) (Crammer and Singer, 2003) for learning weights and employed cutting plane inference (Riedel, 2008) with integer linear programming as its base solver for inference at test time as well as during the MIRA online learning process.

2.2 MLN-based GN methods

Our MLN-based GN system is designed to identify gene mentions in a given biomedical article and normalize them to EntrezGene IDs. First, a gene mention recognizer identifies gene mention boundaries in a given article. The system then looks up each gene candidate in a lexicon of gene names and IDs compiled from EntrezGene and normalizes it to an EntrezGene ID using our MLN-based method. We define three predicates, *isNormalizedTo*, *ShouldBeNormalized* and *isCoreference*, to capture the concept of GN, FP-filter and co-reference, respectively.

Before entering into subsections, we introduce some basic logic symbols used in this work to avoid misunderstanding. The Boolean operations of logical conjunction, disjunction and negation are denoted by $\wedge$, $\vee$ and $\neg$, respectively. The symbol, '$\Rightarrow$', means implies; $A \Rightarrow B$ means if A is true, then B is also true; if A is false then nothing is said about B. The symbol, '$\exists$', is an existential quantification while '$\exists!$' is a uniqueness quantification. Note that $\exists x.P(x)$ means there is at least one x such that $P(x)$ is true. But $\exists! x.P(x)$ means there is exactly one x such that $P(x)$ is true.

2.3 Formulae for GN constraints

For GN, we use the predicate *isNormalizedTo*(i, id) to represent that the i-th gene mention is normalized to id. Note that, in our formulae, we refer to a gene mention by its order in the article (e.g. the i-th gene mention) for several reasons. First, not all names can be found in the training data. Secondly, even if two gene mentions have the same name string, they may normalize to different IDs. Lastly, this allows us to model the order information and dependency among all gene mentions for disambiguation.

The most general assumption is that if a gene mention is mapped to only one ID, it should be normalized to that ID. This is defined as:

Formula 1: $\exists! id.isCandidateOf(id,\ i) \Rightarrow isNormalizedTo(i,\ id)$

where *isCandidateOf*(id, i) represents that the ID candidate id is a candidate of the i-th gene mention.

Because the objective of GN task is to discover a unique ID for each gene mention, we must define a formula to ensure that the constraint is satisfied. We use the following formula to ensure that each gene mention is normalized to only one ID:

Formula 2: $isNormalizedTo\,(i,\ id_1) \wedge id_1 \neq id_2 \Rightarrow \neg isNormalizedTo\,(i,\ id_2)$

The formula is a hard constraint that must always hold, whereas Formula 1 is soft and can be violated. The weights of soft constraints can be learned from training data.

Finally, if a gene mention has two or more candidate IDs, we must determine which is more appropriate through the disambiguation processing. In the next subsection, we discuss the disambiguation formulae defined for *isNormalizedTo*. We refer to *isNormalizedTo* and the other two predicates, *ShouldBeNormalized* and *isCoreference*, for FP-filter and co-reference as 'hidden' because we need to infer them at test time. In contrast, predicates defined in the following sections are considered 'observed', because they are known in advance.

2.4 Formulae for disambiguation

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 1. Main observed predicates for GN

Profile based	*isCandidateOf*(id, i): id is a candidate ID of the i-th gene mention. *hasChromosomeInfo*(i, id, sd): 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. *isPPIPartner*(id_1, id_2): id_1 and id_2 are an interaction pair. *hasPPIPartnerRank*(i, id, 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. *PPIKeyword*(w): the word w is a protein-protein interaction keyword.
Non-profile based	*hasPrecedingWord*(i, w, l): the i-th gene mention has a preceding word w in the range l. *hasUnigramBetween*(i,j, u): there is an unigram u between the i-th and the j-th gene mentions. *hasWord*(w): the abstract contain a word w.

Please refer to the Supplementary 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 UBQLN3 gene was mapped to the 11*p*15 region of chromosome 11.'

The mention UBQLN3 must be normalized to the EntrezGene ID:50613 because 50613's chromosome location, 11*p*15, is found in the same sentence. The formula describing the relation of *hasChromosomeInfo* and *isNormalizedTo* is defined as follows:

$$hasChromosomeInfo\,(i,\ id,\ +sd) \Rightarrow isNormalizedTo(i,\ id)$$

Here, we can see that there is an additional parameter, $+sd$, in the predicate *hasChromosomeInfo*. sd, 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, sd is 1. The '+' notation in the above formula indicates that the MLN must learn a separate weight for each grounded variable (sd). 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{1,2}[pq]\d{1,2}\.?\d{1,2}?-?[pq]?\d{1,2}?\.?\d{1,2}?' 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 i 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*(i, id, r) to represent this concept. The formula defining the relationship between *hasPPIPartnerRank* and *isNormalizedTo* is:

$$\exists! id.hasPPIPartnerRank\,(i,\ id, 1) \wedge hasWord\,(w) \wedge PPIKeyword\,(w)$$
$$\Rightarrow isNormalizedTo\,(i,\ id)$$

One can see that there are two predicates in this formula that check if the article contains protein–protein interaction (PPI) keywords[1] 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 terms or tissue terms 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_2 if another gene mention *i* has been normalized to id_1 and id_1 forms an interaction with id_2:

Formula 3: $hasWord\,(w) \wedge PPIKeyword\,(w) \wedge isNormalizedTo(i,\ id_1) \wedge$ $\text{is}Candidate Of\,(id_2,\ j) \wedge isPPIPartner(id_1,\ id_2) \Rightarrow isNormalizedTo\ (j,\ id_2)$

2.4.2 Non-profilebased 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:

$$hasPrecedingWord\ (i,\ \text{"("},\ 1) \wedge hasFollowingWord\ (i,\ \text{")"},\ 1)$$

$$\wedge \exists! u.hasUnigramBetween\ (i,j,\ u) \wedge u = \text{"("} \wedge \exists! id.isCandidateOf\,(id,\ i)$$

$$\Rightarrow isNormalizedTo(\,j,\ id)$$

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

Salience formula: $i < j \wedge isNormalizedTo(i, id) \wedge isCandidateOf\ (id,\ j) \Rightarrow$ $isNormalizedTo(\,j, id)$

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.

Formula 4: $isNormalizedTo(i,\ id) \Rightarrow ShouldBeNormalized\ (i)$

Note that the formula is equivalent to

$$\neg isNormalizedTo(i,\ id) \vee ShouldBeNormalized\ (i)$$

which models the FP-filtering stage decision determined by traditional separate-stage GN systems: the recognized gene mention *i* does not have to be normalized to the identifier *id*; however, the *id* cannot be assigned to the entity *i* that has not been proposed as a potential gene mention that should be normalized. The formula is a hard constraint.

For GN, FPs can be classified into two types: those that do not belong to any entity class, and those that belong to classes that are not the curation target, e.g. DNA polymerases, protein families or in a specific organism that is not considered. In our model, for the *i*-th entity, if the possible world *ShouldBeNormalized* (*i*) is false, the entity is considered as an FP.

The first formula containing *ShouldBeNormalized* is associated with different weights by considering the grounded gene name *n*:

$$hasGeneName(i,\ +n) \Rightarrow ShouldBeNormalized\ (i)$$

Table 2. Main observed predicates for FP-filtering

hasGeneName(*i*, *ws*): the name of the *i*-th gene mention is *ws*
hasFirstWord(*i*, *w*): the first word of the *i*-th gene mention is *w*
SpeciesTerm(*w*): the word *w* is a species term.
Blacklisted(*n*): the word sequence *n* is blacklisted.
containsMoreSpecificMentions(*i*): the *i*-th gene mention collocates with more specific gene mentions in the current context.
AllUpperCases(*ws*): the word sequence *ws* are all uppercase.

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:

Formula 5:

$$hasFirstWord(i,\ +w) \wedge SpeciesTerm(+w) \Rightarrow ShouldBeNormalized\ (i)$$

implies that a certain gene mention *i*'s suitability for normalizing depends on whether or not *i*'s first word is a certain species keyword. Again, the '+' notation in the above formula indicates that the MLN must learn a separate weight for each gene name *n*

Weeber *et al.* (2003) found that abbreviations of gene mentions in biomedical language are highly ambiguous with other biomedical terms, and many of the occurrences in MEDLINE abstracts are not really related to the corresponding genes. We use the following formula to combine the concept with the blacklist.

$$hasGeneName(i,\ n) \wedge AllUpperCases(n) \wedge$$

$$Blacklisted(n) \wedge hasPrecedingWord\ (i,\ \text{"("},\ 1) \wedge$$

$$\exists! u.hasUnigramBetween\ (i,j,\ u)$$

$$\wedge u = \text{"("} \Rightarrow \neg ShouldBeNormalized(\,j,\ id)$$

The formula states that if an all-capitalized mention *i* in parentheses is blacklisted (i.e. such mention is assumed to be an abbreviation), its preceding mention *j* is very likely not to be normalized.

In addition, we use predicate *containsMoreSpecificMentions*(*i*) to indicate that *i* collocates with more specific gene mentions in the same abstract[2] If the grounded value is true, *i* should not be normalized to any ID. The actual formula is defined as follows:

$$containsMoreSpecificMentions\ (i) \Rightarrow \neg ShouldbeNormalized\ (i)$$

Consider the IL-1 family as an example to clarify this formula. The IL1 family includes IL1 alpha, beta, IL1ra, etc. In the EntrezGene database, there is no ID for the family mention (IL1), but it does include the IDs for IL1alpha/beta and IL1ra. Using the above formula in the sentence, '…<entity id=7850>type II IL-1 receptor</entity> can bind all three forms of IL-1 (<entity id=3552>IL-1 alpha</entity>, <entity id=3553>IL-1 beta</entity> and <entity id=3557>IL-1ra</entity>)', the mention 'IL-1' will not be normalized, but following gene mentions, including IL-1 alpha, beta and IL-1ra, will be.

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 *isCoreference* (*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

[1] The PPI keywords are collected from Plake et al.'s work (2005).

[2] The regular expression, '\b({0}\s? {1}|type[\s-]?{2} {0}|{0} [\s-]?\d+|{0}\s?{2})\b', is used to check the specificity; {0}is the gene mention, {1} denotes the Roman letters and {2} denotes the Roman numbers.

Table 3. Observed predicates and formulae for co-reference resolution

Predicate $isAliasOf\ (i,\ j\,)$: the i-th gene mention is listed as an alias of the j-th gene mention.
$Distance(i,\ j,\ dis)$: the distance between the i-th and the j-th gene mention is dis.
$InAppositionTo(i,\ j)$: the i-th gene mention is in apposition to the j-th gene mention.

String match feature:
$hasGeneName\ (i,\ n\,) \wedge hasGeneName(\,j,\ n) \Rightarrow isCoreference(i,\ j)$
Alias feature:
$isAliasOf\ (i,\ j\,) \wedge hasGeneName\ (i,\ n\,) \wedge hasGeneName\ (j,\ n\,) \Rightarrow isCoreference(i,\ j)$
Distance feature: $Distance(i,\ j,\ +g) \Rightarrow isCoreference(i,\ j)$
Appositive feature: $InAppositionTo(i,\ j) \Rightarrow isCoreference(i,\ j)$

Transitivity formula: $isCoreference\ (i,\ j\,) \wedge isNormalizedTo\ (i,\ id_1\,) \wedge \neg\exists id_2.isNormalizedTo\ (j,\ id_2\,) \Rightarrow isNormalizedTo\ (j,\ id_1\,)$
to express the transitivity concept that if the i-th gene mentions and the j-th are co-reference, and i is normalized to id and j has not been normalized, then j should be also normalized to id. There are several co-reference works in general domains (Ng, 2005; Poon and Domingos, 2008; Soon *et al.*, 2001). In this work, we implement a subset of features presented by Soon *et al.* (2001). Table 3 lists the additional observed predicates defined for *isCoreference*, the formula definition and its corresponding feature name defined by Soon *et al.* Please refer to their work for details of the feature descriptions.

In addition, in this work, we only consider gene mentions as potential co-reference candidates. Following the same concept described for Formula 2, we add the following structural constraint to ensure that whenever i and j are co-reference, they must be gene mentions should be normalized.

$$isCoreference(i,\ j) \Rightarrow ShouldBeNormalized\ (i\,) \wedge ShouldBeNormalized\ (j\,)$$

Finally, we must ignore the first order logic *unique names assumption* (Russell and Norvig, 1995)—i.e. different constants always refer to different objects in the domain. This assumption can be removed by employing the following hard constraint formulae in our models:

Reflexivity: $\forall i.isCoreference(i,\ i)$

Symmetry: $\forall i,\ j.isCoreference(i,\ j) \Rightarrow isCoreference(\,j,\ i)$

Transitivity: $\forall i,\ j,k.isCoreference(i,\ j) \wedge isCoreference(\,j,\ k) \Rightarrow isCoreference(i,\ k)$

3 RESULTS

3.1 Experimental setup

3.1.1 Evaluation schemes We use the standard recall, precision and F-measure metrics (RPF) 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 ID list for the article. The RPF 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 IE resolution. In contrast to the first metric, the RPF 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/FP/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>serum LH</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 (Lu 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.[3]

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.

[3] 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 publically 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 Model$_1$ and Model$_2$. In ME modeling, we formulated both the FP-filtering and the co-reference resolution tasks as classification tasks. Model$_1$ used the features equivalent to the formulae described in Section 2.5 (FP-filtering). Model$_2$ 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¢wthe 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, Model$_1$ 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, Model$_2$ 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 Model$_2$, the algorithm chose the ID with the highest confidence from the co-reference chain, and then assigned it to *i*

Table 4. The results on the test set using instance-based criterion

Configuration		P	R	F	Diff
No disambiguation/**MLN-based: 2.3**		80.7	56.3	66.3	0
(a)	**MLN-based/2.3+2.4**	73.8	64.3	**68.7**	+2.4
(b)	System 1	72.8	63.9	68.1	+1.8
(c)	System 2	79.3	58.4	67.3	+1.0
(d)	**MLN-based/ (a)+2.5**	79.2	62.8	**70.0**	+3.7
(e)	System 1 +Model$_1$	73.5	63.9	68.4	+2.1
(f)	System 2 +Model$_1$	80.2	58.4	67.6	+1.3
(g)	**MLN-based/ (d)+2.6**	77.8	65.3	**71.0**	+4.7
(h)	System 1 +Model$_1$ +Model$_2$	73.2	64.7	68.7	+2.4
(i)	System 2 +Model$_1$ +Model$_2$	79.9	58.9	67.8	+1.5

The MLN-based approaches are highlighted in bold.

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 Model$_1$ 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

Table 5. Results derived on the test set using the article-wide evaluation

Configuration		P	R	F	Diff
No disambiguation/**MLN-based: 2.3**		77.3	71.4	74.2	0
(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	+8.8
(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

Table 6. The efforts of salience and transitivity on the test set

Configuration		P	R	F	Diff
(a)	No disambiguation/**MLN-based: 2.3**	80.7	56.3	66.3	0
(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)	(a) + Transitivity +GOLD	80.6	67.5	73.5	+7.2
(f)	(e) + Salience	82.5	67.5	74.2	+7.9

'GOLD' in configuration (e) refers to using the gold co-reference annotations.

disambiguation formulae boosts both instance-based and article-wide evaluation by apparent large margin (3.7 and 10.3%). By adding the FP-filtering and co-reference formula/features, our MLN-based GN system does better than the compared separate-stage methods under both two evaluation criteria.

4 DISCUSSION

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 co-reference 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 co-reference 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 can be used to improve the fine-grained IE GN's performance.

The results also show one advantage of employing MLN in our GN modeling: MLN allow us to easily model arbitrary longer range dependencies as expressed by the Salience, Transitivity and Formula 3. However, it is difficult to directly model such dependencies using machine learning algorithms like ME. We believe that a MLN model is preferable in this case.

We also observe that adding these formulae (Salience and Transitivity) do not have significant efforts 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.

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: '**ABCB9 protein** appears to be most highly 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 'Arl**2/3**', '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.

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):

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://cbioc.eas.asu.edu/banner/) 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: <entityid=3082>HGF</entity> activates <entityid=3082>HGF</entity>.

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 S.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.* (2009), 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 Model$_1$ 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 the reason 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 Model$_2$ with ME Model$_1$ 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.

6 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 G.N. 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 IIS-50-23.

Conflict of Interest: none declared.

REFERENCES

Aha,D.W. and Bankert,R.L. (1995) A comparative evaluation of sequential feature selection algorithms. In Fisher,D. and Lenz,H-J. (eds) *Learning from Data: Artificial Intelligence and Statistics V*, Springer, pp. 199–206.

Baumgartner,W.A. Jr *et al.* (2007) An integrated approach to concept recognition in biomedical text. In *Proceedings of the Second BioCreative Challenge Evaluation Workshop, CNIO (Centro Nacional de Investigaciones Oncologicas)*, pp. 257–271.

Crammer,K. and Singer, Y. (2003) Ultraconservative online algorithms for multiclass problems. *J. Mach. Learn. Res.*, **3**, 951–991.

Crim,J. *et al.* (2005) Automatically annotating documents with normalized gene lists. *BMC Bioinformatics*, **6**, S13.

Dai,H.-J. *et al.* (2010) Multistage gene normalization and SVM-based ranking for protein interactor extraction in full-text articles. *IEEE Trans. Comput. Biol. Bioinformatics*, **7**, 412–420.

Finkel,J. *et al.* (2005) Exploring the boundaries: gene and protein identification in biomedical text. *BMC Bioinformatics*, **6**, S5.

Grosz,B. *et al.* (1995) Centering: a framework for modeling the local coherence of discourse. *Comput. Ling.*, **21**, 203–225.

Hakenberg,J. *et al.* (2008) Inter-species normalization of gene mentions with GNAT. *Bioinformatics*, **24**, 126–132.

Khalid,M. *et al.* (2008) The impact of named entity normalization on information retrieval for question answering. *Adv. Informat. Retr.*, **4956**, 705–710.

Lai,P.-T. *et al.* (2009) Using contextual information to clarify gene normalization ambiguity. *In IEEE International Conference on Information Reuse and Integration (IEEE IRI 2009)*. IEEE Press, Las Vegas, USA, pp. 1–5.

Li,Y. *et al.* (2009) Incorporating rich background knowledge for gene named entity classification and recognition. *BMC Bioinformatics*, **10**, 223.

Lu,Z. *et al.* (2011) The gene normalization task in BioCreative III. *BMC Bioinformatics*, Available at http://www.ncbi.nlm.nih.gov/CBBresearch/Lu/publications.html.

Meza-Ruiz,I. and Riedel,S. (2009) Jointly identifying predicates, arguments and senses using Markov logic. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Boulder, CO, pp. 155–163.

Morgan,A. *et al.* (2008) Overview of BioCreative II gene normalization. *Genome Biol.*, **9**, S3.

Neves,M. *et al.* (2010) Moara: a Java library for extracting and normalizing gene and protein mentions. *BMC Bioinformatics*, **11**, 157.

Ng,V. (2005) Machine learning for coreference resolution: from local classification to global ranking. In *Proceedings of the 43rd Annual Meeting of the Asssociation for Computational Linguistics (ACL'05)*, Association for Computational Linguistics, pp. 157–164.

Plake,C. *et al.* (2005) Optimizing syntax patterns for discovering protein-protein interactions. In *Proceedings of the 2005 Association for Computing Machinery symposium on Applied computing*. ACM, Santa Fe, New Mexico.

Poon,H. and Domingos,P. (2008) Joint unsupervised coreference resolution with Markov Logic. In *Proceedings of the 2008 Conference on Empirical Methods* *in Natural Language Processing*. Association for Computational Linguistics, Honolulu, pp. 649–658.

Richardson,M. and Domingos,P. (2006) Markov logic networks. *Mach. Learn.*, **62**, 107–136.

Riedel,S. (2008) Improving the accuracy and efficiency of map inference for markov logic. In *Proceedings of the Association for Uncertainty in Artificial Intelligence's (UAI'08)*. AUAI Press.

Russell,S. and Norvig,P. (1995) *Artificial Intelligence: a Modern Approach*. Prentice Hall, Englewood Cliffs, NJ, London.

Smith,L. *et al.* (2008) Overview of BioCreative II gene mention recognition. *Genome Biol.*, **9**, S2.

Soon,W.M. *et al.* (2001) A machine learning approach to coreference resolution of noun phrases. *Comput. Ling.*, **27**, 521–544.

Subramaniam,L.V. *et al.* (2003) Information extraction from biomedical literature: methodology, evaluation and an application. In *Proceedings of the twelfth international conference on Information and knowledge management*. ACM, New Orleans, LA, USA, pp. 410–417.

Tsai,R.T.-H. *et al.* (2006) Various criteria in the evaluation of biomedical named entity recognition. *BMC Bioinformatics*, **7**, 14.

Tsuruoka,Y. *et al.* (2007) Learning string similarity measures for gene/protein name dictionary look-up using logistic regression. *Bioinformatics*, **23**, 2768.

Wang,X. *et al.* (2010) Disambiguating the species of biomedical named entities using natural language parsers. *Bioinformatics*, **26**, 661–667.

Weeber,M. *et al.* (2003) Ambiguity of Human Gene Symbols in LocusLink and MEDLINE: creating an inventory and a disambiguation test collection. In *Proceedings of the American Medical Informatics Association Symposium*. pp. 704–708.

Xu,H. *et al.* (2007) Gene symbol disambiguation using knowledge-based profiles. *Bioinformatics*, **23**, 1015–1022.

Yoshikawa,K. *et al.* (2010) Coreference Based Event-Argument Relation Extraction on Biomedical Text. In *Proceedings of the Fourth Symposium on Semantic Mining in Biomedicine (SMBM 2010)*. European Bioinformatics Institute, Hinxton, Cambridgeshire, UK.