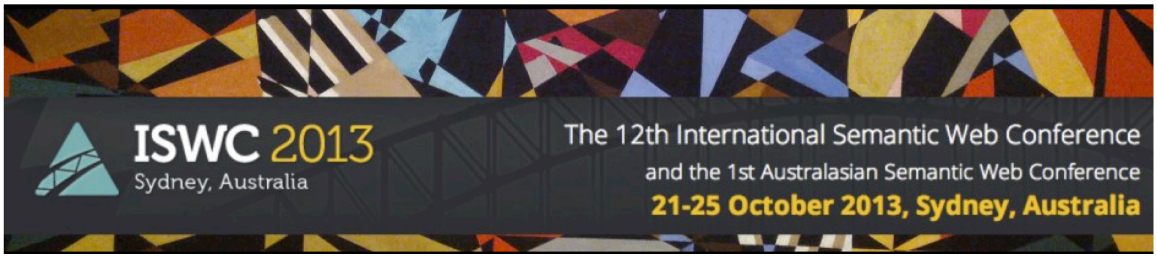
CrowdSem 2013

The Confluence of Crowdsourcing and Semantic Web


ISWC 2013
Sydney, Australia
The 12th International Semantic Web Conference
and the 1st Australasian Semantic Web Conference
21-25 October 2013, Sydney, Australia

Proceedings
of the
First International Workshop
on

Crowdsourcing the Semantic Web

held in conjunction with
the 12th International Semantic Web Conference

Sydney, Australia

Editors:
Maribel Acosta
Lora Aroyo
Abraham Bernstein
Jens Lehmann
Natasha Noy
Elena Simperl

Preface

This volume contains the papers presented at the 1st International Workshop on ”Crowdsourcing the Semantic Web” that was held in conjunction with the 12th International Semantic Web Conference (ISWC 2013), 21-25 October 2013, in Sydney, Australia. This interactive workshop takes stock of the emergent work and chart the research agenda with interactive sessions to brainstorm ideas and potential applications of collective intelligence to solving AI hard semantic web problems.

There were 12 submissions. Each submission was reviewed by at least 2, and on the average 3, program committee members. The committee decided to accept 9 papers.

Our special thanks goes to the reviewers who diligently reviewed the papers within this volume.

September 3, 2013

Maribel Acosta
Lora Aroyo
Abraham Bernstein
Jens Lehmann
Natasha Noy
Elena Simperl

Table of Contents

Full Papers

Experiment Papers

Position Papers

Program Committee

Maribel Acosta
Lora Aroyo
Soren Auer
Abraham Bernstein
Irene Celino
Oscar Corcho
Philippe Cudre-Mauroux
Roberta Cuel
Dave de Roure
Yolanda Gil
Tudor Groza
Michiel Hildebrand
Haym Hirsh
Aidan Hogan
Mark Klein
Jens Lehmann
Patrick Minder
Dunja Mladenic
Jonathan Mortensen
Mathias Niepert
Barry Norton
Natasha Noy
Harald Sack
Cristina Sarasua
Elena Simperl
Jamie Taylor
Amrapali Zaveri
Jun Zhao

Additional Reviewers

Demartini, Gianluca
Waitelonis, Joerg

Crowdsourced Semantics with Semantic Tagging: “Don’t just tag it, LexiTag it!”

Csaba Veres
Institute for Information and Media Science,
University in Bergen, Norway
Csaba.Veres@infomedia.uib.no

Abstract. Free form tagging was one of the most useful contributions of “Web2.0” toward the problem of content management and discovery on the web. Semantic tagging is a more recent but much less successful innovation borne of frustration at the limitations of free form tagging. In this paper we present LexiTags, a new platform designed to help realize the potential of semantic tagging for content management, and as a tool for crowdsourcing semantic metadata. We describe the operation of the LexiTags semantic bookmarking service, and present results from tools that exploit the semantic tags. These tools show that crowdsourcing can be used to model the taxonomy of an information space, and to semantically annotate resources within the space.

Keywords. crowdsourcing, metadata, bookmarking, tagging, semantic tags

1 Introduction

The emergence of "Web2.0"[1] brought a number of innovations which changed the way people interact with information on the World Wide Web. The new paradigms made it easy for anyone to contribute content rather just consume. One of the early success stories was *social tagging*, which gave rise to *folksonomies*[2] as a way to organise and find information on the Web through emergent shared vocabularies developed by the users themselves. Social tagging for content management and discovery became very popular in commercial services like the photo sharing site flickr.com and the bookmarking site delicious.com. These successes prompted some commentators to declare victory of user driven content tagging over the “overly complex” technologies of the semantic web. Perhaps most famously, in a web post entitled “Ontology is Overrated: Categories, Links, and Tags” Clay Shirky argued that any technology based on hierarchical classification (including ontologies) was doomed to fail when applied to the world of electronic resources [1]. Instead, simple naive tagging opened the door to crowdsourced content management, where dynamic user contributed metadata in a flat tag space offered a breakthrough in findability.

However, researchers and information architects soon began to point out the limitations of unconstrained tagging for enhancing information findability. [2] identified a number of problems with tagging, which can limit its effective usefulness. Among the problems were tag ambiguity (e.g. apple - fruit vs. apple - company), idiosyncratic

[1] T. O'Reilly. What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html, 2005.

[2] http://iainstitute.org/news/000464.php#000464

treatment of multi word tags (e.g. vertigovideostillsbbc, design/css), synonyms (e.g. Mac, Macintosh, Apple), the use of acronyms and other terms as synonymous terms (e.g. NY, NYC, Big Apple), and of course mis spelled and idiosyncratic made up tags. These factors limit the use of tags in large scale information management. For example [3] discuss limitations of searching with tags, which is necessarily based on syntactic matching since there are no semantic links between individual tags. Thus, searching with "NYC" will not guarantee that results tagged with "NY" will be retrieved.

Semantic tags or *rich tags* as they are known in the context of social tagging, emerged as a way to impose consistent and refined meanings to user tags [4]. The two best known semantic tagging sites were Faviki[3] and Zigtag[4] (the latter now appears to be defunct). Each site expected users to use tags from a large collection of provided terms. Faviki used WikiPedia identifiers, while Zigtag used a "semantic dictionary". Both sites also allowed tags which were not in their initial knowledge base. In the case of Faviki, users can link undefined tags to a web page which best represents the tag. The web page is located by a simple web search. Zigtag allowed the use of undefined tags, with the expectation that users would later return and provide definitions of the tags. However a large proportion of tags remained without definition, leading to a mishmash of defined and undefined tags.

LexiTags [5] was initially developed for similar reasons, as a tool for content management with rich tags. But as a semantic application it also had higher aspirations, to provide a platform and a set of front end tools designed to crowdsource the semantic web. By providing intuitive access points (APIs) to user generated metadata with semantic tags, the platform was expected to outgrow its initial purpose and provide novel new benefits for its users while at the same time generating semantic metadata for general consumption. The intuition is that users should have systems which behave as Web2.0 at the point of insertion, yet as Semantic Web at the point of retrieval. In this paper we present the core LexiTags system, and describe some tools we have developed to capitalise on the crowdsourced semantic metadata.

2. LexiTags

LexiTags[5] is an acronym for "Lexical Tags", from the fact that the tags are primarily lexical items, or natural language dictionary words. They are disambiguated through the use of an interface which presents the user with a set of choices from WordNet, an electronic lexical database [6]. The main content bearing units in WordNet are *synsets*, which are represented by contextually synonymous word meanings grouped in a single entry. The word *couch* for example is represented by the synset {*sofa, couch, lounge*}. But *couch* is of course an ambiguous word whose alternate meaning appears is a second synset {*frame, redact, cast, put, couch*} as in "Let me couch that statement …. ". There is therefore no ambiguity in WordNet because every synset represents a unique meaning for a word string. LexiTags presents such synsets, and a short gloss, to help users chose their intended meaning.

The use of synsets as tags can combine precise definitions without the need to adopt a set of idiosyncratic keywords. Mappings can be set up between synsets and any other vocabulary, enabling specific keyword markup through a natural language interface.

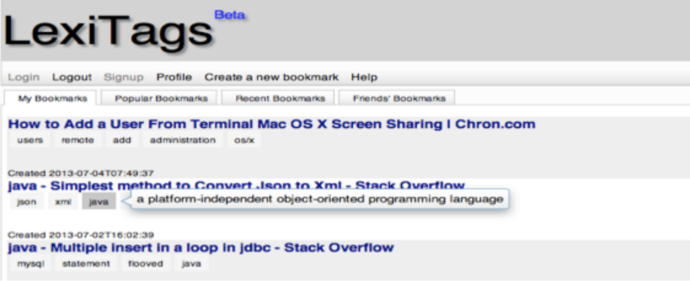
[3] http://www.faviki.com/pages/welcome/

[4] http://zigtag.com

[5] http://lexitags.dyndns.org/ui/webgui/

The success of mapping efforts can of course vary. While highly technical ontologies could prove difficult, lightweight ontologies and taxonomies like schema.org are not problematical, and in section 3.2 we will see a tool that makes use of exactly such a mapping.

Fig. 1 shows a detail of the main interface of LexiTags, which is mainly a simple list of URLs that have been bookmarked. Clicking on the URL opens a new window with the web site. The user tags appear below the URL. Hovering the mouse over these tags pops up a definition which clarifies the precise sense of that tag. Clicking on a tag will open a new tab which shows bookmarks tagged with the same tag sense.



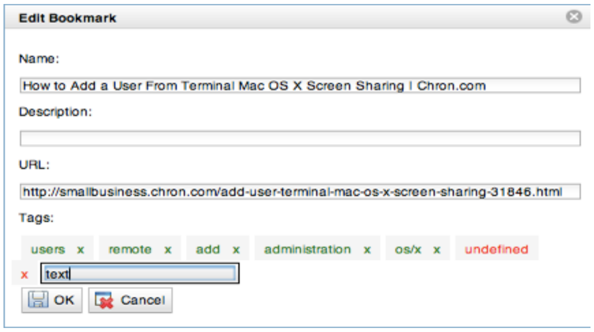
How to Add a User From Terminal Mac OS X Screen Sharing I Chron.com
users remote add administration os/x

Created 2013-07-04T07:49:37
java - Simplest method to Convert Json to Xml - Stack Overflow
json xml java
a platform-independent object-oriented programming language

Created 2013-07-02T16:02:39
java - Multiple insert in a loop in jdbc - Stack Overflow
mysql statement flooved java

Fig. 1. The main LexiTags interface

Tags for a bookmark are entered freely in the text box near the bottom of the "Edit Bookmark" window (fig. 2) which is popped up through the use of a bookmarklet. The user types one tag at a time into the text box and presses enter which puts each tag in a list above the text box, initially in red colour. Users can simply enter tags as they like, as in most other tagging sites. Finally, users click on each undefined tag to add disambiguation through the final "Editing the tag …" window, shown in figure 3.



Name:
How to Add a User From Terminal Mac OS X Screen Sharing I Chron.com

Description:

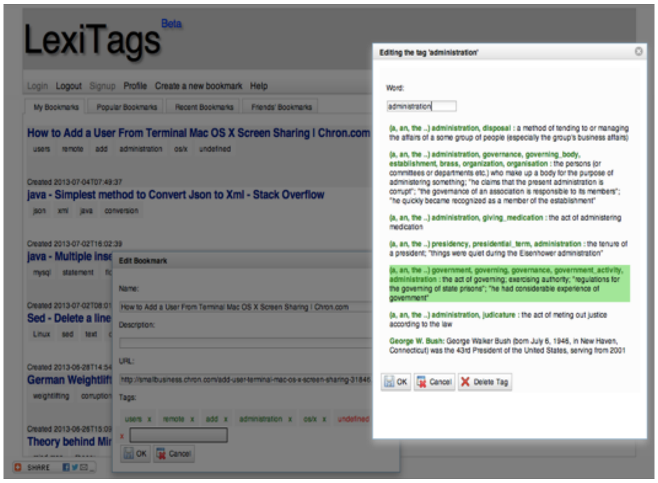
URL:
http://smallbusiness.chron.com/add-user-terminal-mac-os-x-screen-sharing-31846.html

Tags:
users x remote x add x administration x os/x x undefined
x text

Fig. 2. Window for adding and editing tags

The "Edit tag .." window shows the possible interpretations from WordNet, or DBPedia if WordNet does not have an entry for the tag word. This is the case most often when the tag is the name of a company or a person or a new technology. DBPedia also includes a large number of common abbreviations, such as "NYC". In addition, DBPedia defines mappings to WordNet synsets for many concepts, which helps fill gaps in WordNet. Unfortunately the coverage is not complete, so "NYC" for example is not linked to the synset for New York in WordNet. Users must select the sense that best matches their intent. The choices are ordered by word frequency, and our experience suggests that the intended sense is amongst the first two or three senses. Since the main point of tagging is for future retrieval, it would make sense if people tended to avoid words in their obscure, low frequency senses. However this is just conjecture at the moment, and we are investigating other methods for optimal ordering. One approach is to weight the rankings according to the aggregate distance of each candidate sense to a context tag provided by already disambiguated tags. Another approach is to personalise the rankings so that each user's own tagging history influences the ranking of the candidate senses.

As each tag is disambiguated by the user, it turns green. Any tag left disambiguated is deleted when the user presses the "OK" button, so every tag in the LexiTags platform is a disambiguated, defined term.



How to Add a User From Terminal Mac OS X Screen Sharing | Chron.com
users remote add administration os/x undefined

Created 2013-07-04T07:49:37
java - Simplest method to Convert Json to Xml - Stack Overflow
json xml java conversion

Created 2013-07-02T16:02:39
java - Multiple inse
mysql statement

Created 2013-07-02T08:01
Sed - Delete a line
Linux sed text

Created 2013-06-28T14:54
German Weightlift
weightlifting corruption

Created 2013-06-28T15:09
Theory behind Mir

Edit Bookmark

Name:
How to Add a User From Terminal Mac OS X Screen Sharing | Chron.com

Description:

URL:
http://smallbusiness.chron.com/add-user-terminal-mac-os-x-screen-sharing-31846

Tags:
users x remote x add x administration x os/x x undefined

Editing the tag 'administration'

Word:
administration

(a, an, the ...) administration, disposal : a method of tending to or managing the affairs of a some group of people (especially the group's business affairs)

(a, an, the ...) administration, governance, governing_body, establishment, brass, organization, organisation : the persons (or committees or departments etc.) who make up a body for the purpose of administering something; "he claims that the present administration is corrupt"; "the governance of an association is responsible to its members"; "he quickly became recognized as a member of the establishment"

(a, an, the ...) administration, giving_medication : the act of administering medication

(a, an, the ...) presidency, presidential_term, administration : the tenure of a president; "things were quiet during the Eisenhower administration"

(a, an, the ...) government, governing, governance, government_activity, administration : the act of governing; exercising authority; "regulations for the governing of state prisons"; "he had considerable experience of government"

(a, an, the ...) administration, judicature : the act of meting out justice according to the law

George W. Bush: George Walker Bush (born July 6, 1946, in New Haven, Connecticut) was the 43rd President of the United States, serving from 2001

Fig. 3. Disambiguation window

3. The Lexitags ecosystem

If LexiTags were just a bookmarking service with rich tags, there would be little to differentiate it from Zigtags. But the idea was to use the bookmarked sites and their tags as a starting point for a set of tools that extracted value from the tags. As such, LexiTags should be seen as a platform to expose crowdsourced semantic metadata to clients, both for creation and consumption of metadata. In terms of content creation clients, we are developing an iPhone app for tagging photographs with LexiTags, as described in [5]. Due to space limitations we are unable to discuss alternative input applications, but instead describe two applications for consuming the metadata. One creates a content taxonomy, the other produces metadata for the web.

3.1 Content taxonomy

[7] discuss SynsetTagger[6], which was developed to consume LexiTags tags automatically, but has thus far only been demonstrated in manual mode to create lightweight ontologies from user input. SynsetTagger makes use of select WordNet

[6] http://csabaveres.net/csabaveres.net/Semantic_Apps.html

relations to construct a lightweight ontology by inferring additional nodes from the provided tags. The most important link for nouns is *hyponymy/hypernymy* which are the semantic relations otherwise known as *subordination/superordination*, *subset/superset*, or the IS-A relation. A concept represented by the *synset* {x, x′, . . .} is said to be a *hyponym* of the concept represented by the *synset* {y, y′, . . .} if native speakers of English accept sentences constructed from such frames as ”an x is a (kind of) y” [8], [9]. Another relation used by SynsetTagger is *meronymy*, or the *part/whole* relation. A concept {x,x'….} is a *meronym* of a concept {y, y′, . . .} if "an x is a part of y", and it is a *holonym* if "a y is a part of x". There are several other important relations in WordNet, some of which will be mentioned in the case study.

SynsetTagger works by constructing the hypernym chain and pruning nodes if their information content is trivial, which is determined by counting the outward edges from each node and eliminating each node that falls below a threshold value. For example, fig. 4 shows a taxonomy constructed from the input synsets coloured green. The orange coloured inferred hypernyms will be discarded because the have only a single outgoing edge (or are too close to the top level node), and the clear white ones will be kept. The grey nodes are both inferred and asserted, and will also be kept in the final taxonomy. The nodes that are kept are called *informative subsuming nodes*.

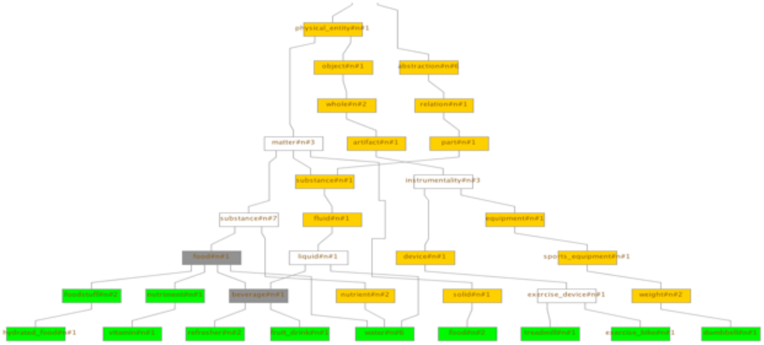


Fig. 4. Complete hypernym chain constructed from input synsets

The tool has a number of user configurable parameters which determine the final selection of nodes. Two important ones are the number of outgoing edges, and the distance from the topmost node. The optimal selection is a matter of trial and error with a given input set. The pruning mechanisms are similar to the "*nearly* automated" approach to deriving category hierarchies for printed text [10], but SynsetTagger differs in that it allows users to adjust the parameters and receive immediate visual feedback about their consequences on the constructed taxonomy. It is intended as an interactive tool to give users a sense of control over their content.

3.2 Metadata for the Web

MaDaME (Meta Data Made Easy) is a tool for embedding semantic metadata into web sites. Its development was spurred by the release of the schema.org initiative, which is a *type* schema meant to be used by web masters to add structured metadata to their content. The incentive for web masters to use the schema is that web sites that contain markup will appear with additional details in search results, which enable people to judge the relevance of the site more accurately and hopefully increase the probability that the site will be visited.

Clearly this is an important development in the effort to crowdsource semantic web content. However, the schema was designed specifically for the use case of search, and both the semantics and the preferred syntax reflect that choice. In terms of semantics, the schema contains some non-traditional concepts to fulfil its intended use. For example there is a general class of *Product* but no general class for *Artifact*. There are also odd property ascriptions from the taxonomy structure, so, for example, *Beach* has *openingHours* and *faxNumber*. In terms of syntax, there is a very strong message that developers should use the relatively new microdata format rather than the more popular RDFa web standard[11]. This is unfortunate because it makes metadata from schema.org incompatible with many other sources of metadata like Facebook's OGP[7].

A strong motivation for MaDaME was to create a tool that would not only help web designers apply schema.org markup to their web sites, but to simultaneously inject ontology terms from other standard sources into their web sites in the RDFa standard. In other words, to maximise the crowdsourcing potential offered by schema.org. This was achieved through mappings between WordNet and schema.org, as well as SUMO [12]. In normal operation users select key words in their web sites, which are then disambiguated using the LexiTags interface. The WordNet synsets are stored, and their most appropriate mapping to schema.org and SUMO computed. These are then inserted into the html source, and made available to the web designer for further refinement.

While MaDaME is currently presented as a standalone tool[8] it can also be used to automatically annotate any web site bookmarked on LexiTags. These annotations can be sent to the maintainers of the web sites with a cover letter explaining the purpose of the markup. Alternatively, the markup could be stored on the LexiTags platform and offered through an API.

4. Results

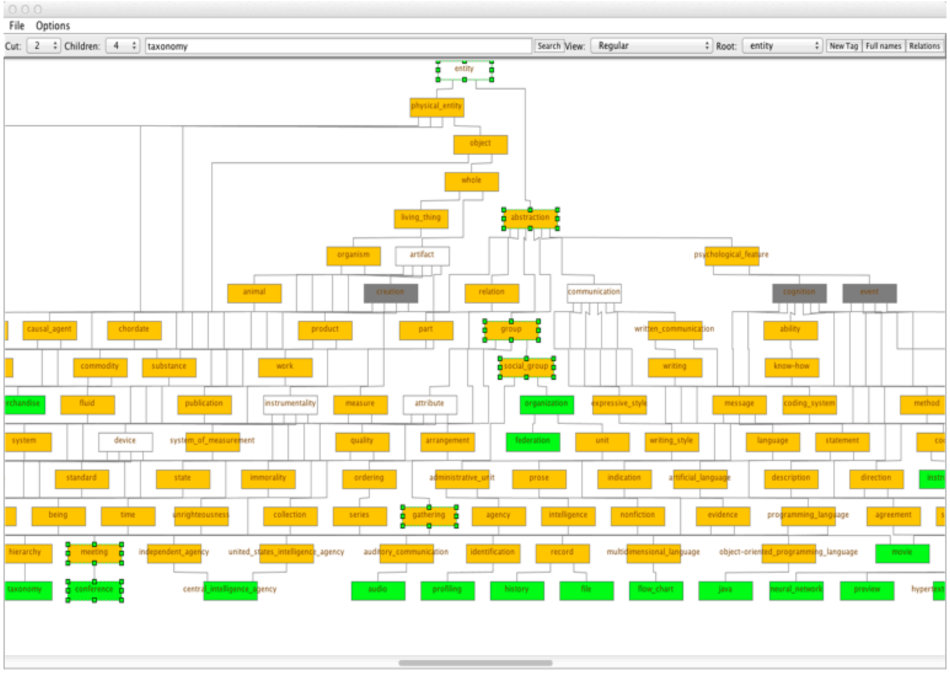
We present a short evaluation of SynsetTagger on a set of approximately 100 bookmarks for a single user on LexiTags, and then the metadata generated for one web site on MaDaME.

Fig. 5 shows a portion of the taxonomy generated from the semantic tags used on the set of 100 bookmarks. The asserted green tags are used to build the hypernym chain from WordNet. If any node is selected in the interface, the set of connected nodes will also be highlighted, making it easier for users to understand the relationships in the taxonomy. For example the asserted tag *conference* can be seen as a kind of *meeting* which in turn is a kind of *gathering*, and so on. The orange nodes are inferred hypernyms, but they will be rejected in the final taxonomy for one of two reasons; a) they have fewer than the specified number of children, or b) they are closer than the selected cut from the *entity* node.The white nodes are the inferred nodes which will be retained in the final export. Users can therefore manipulate the two parameters until the desired level of generality is reached.

We will see that in general a lower number of requisite children and a lower cut will result in fewer nodes. This may be counter intuitive at first, but the rather straightforward explanation is that more and more asserted tags fail to find a suitable subsuming concept when the criterion for retaining the concepts becomes more stringent.

[7] http://ogp.me

[8] http://csaba.dyndns.ws:3000



Cut: 2 Children: 4 taxonomy View: Regular Root: entity

Fig. 5. Taxonomy from approx. 100 bookmarked sites

Fig. 6 shows the hyponyms of *entity* that are retained with "children" set at two and "cut" also set at two. The black arrow mark at the right side of each oval shows that the node can be extended to reveal more children. Every tag except for *weather* found a more general subsuming hypernym when the criterion for subsuming nodes is lax. The tags which do not have a subsuming concept are noteworthy because they represent unusual bookmarks which stand apart from the rest. The subsuming concepts themselves are very general in this example, and their utility for browsing the bookmarked resources is questionable.

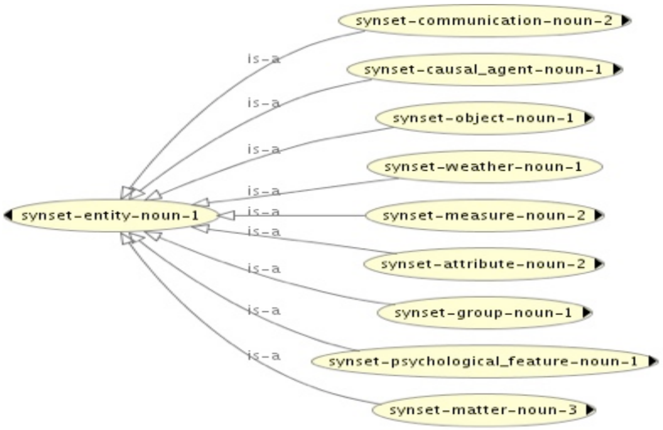


Fig. 6. Inferred hypernyms with lax criteria

Fig. 7 shows the same set of tags with "children" set at 5. Since the criterion for subsuming concepts is much higher, there are many more tags which do not have a subsumer. For example the two tags *fund* and *date* are both a kind of *measure* in fig. 6. But these are the only two kinds of *measure* in the tag set, so with a criterion of 5 children, *measure* is no longer considered as an informative subsuming node. On the other hand the remaining subsuming concepts have differentiated and are now somewhat more specific. For example *group* is replaced with the more specific *organization* which is one of only three kinds of *group* in the tree. So *group* is discarded but *organization* is kept because there are many different kinds of *organization* in the tree. Overall the constructed taxonomy is much more useful for browsing because the general categories are now more informative, and the outliers are not forced into overly general categories.

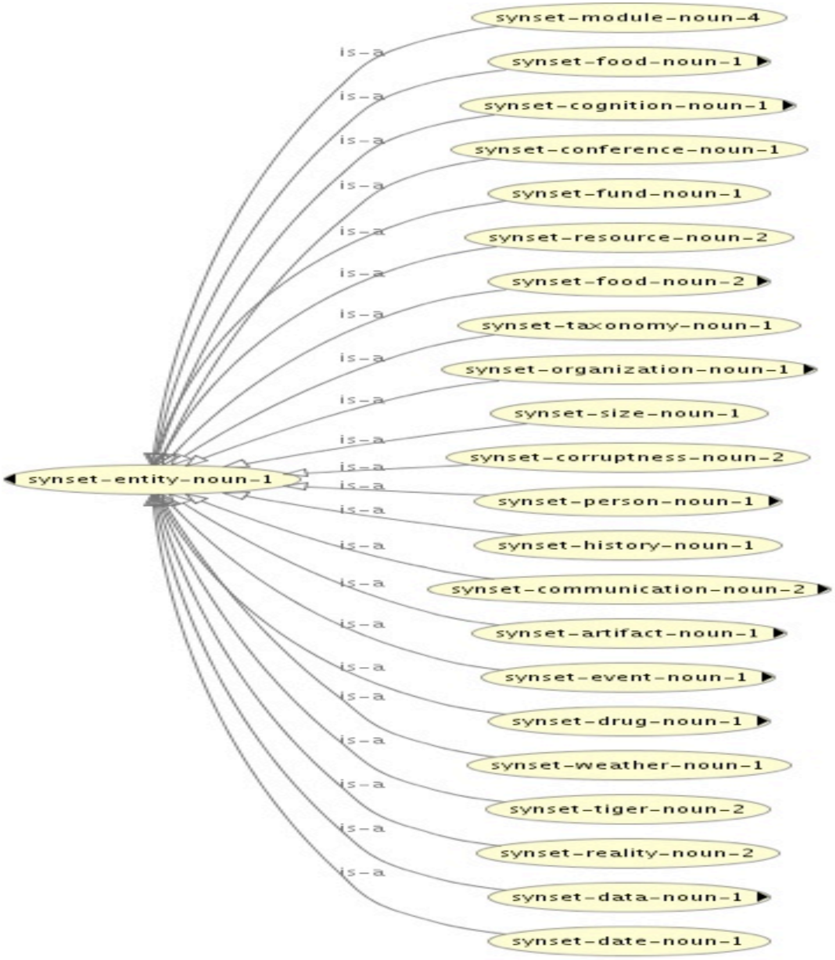


Fig. 7 Inferred hypernyms with stringent criteria

Fig. 8. shows that expanding the *artifact* node reveals some useful sub categories to browse. In general, users can easily configure the parameters to arrive at an optimal taxonomy for their needs.

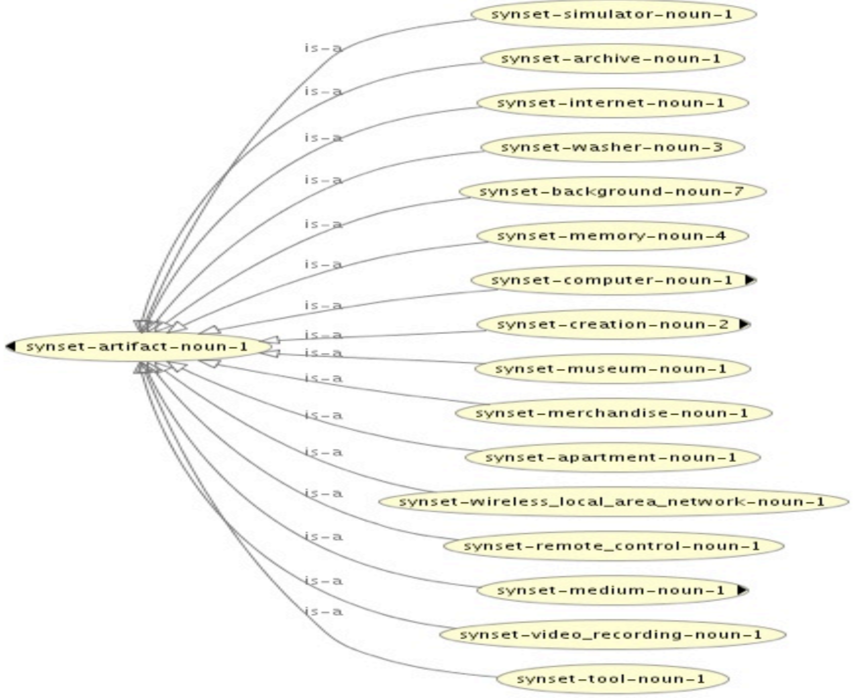


Fig. 8. Different kinds of artefacts in the bookmark set

It is at this point where LexiTags becomes a system that behaves as Web2.0 at the point of insertion, yet as Semantic Web at the point of retrieval. Since the tags have precise definitions they avoid the pitfalls of free form tags like vagueness and ambiguity, and because the defined tags participate in various relations they add value to the asserted tags. They become self organising, with a neat hierarchical structure emerging automatically. Even in this small example of 100 bookmarks and around 400 tags the emerging generalisations form a useful browsing hierarchy. We expect the categorisations to improve with more bookmarks and tags, resulting in a situation where a growing set of bookmarks leads to more organisation rather than complete chaos as seen in traditional free form tagging systems. The users must gain such benefits because they are asked to perform a little more work at the point of insertion. An important point of our work is to enhance LexiTagging services to provide additional benefit to the users, to encourage them to contribute semantic metadata.

The emergent taxonomy can be extended with additional related terms from WordNet. SynsetTagger already has the functionality to add some additional relations to enrich the ontology. Currently these include the two *part-of* relations, and the *domain terms*. For example *movie* has meronyms *episode*, *credits*, *subtitle*, *caption*, *scene*, *shot,* and domain terms *dub, synchronize, film, shoot, take, videotape, tape, reshoot*. These can all be added with the appropriate relations. In addition the synonyms appearing in each synset could also be included. Sometimes this is quite rich, as is the case in our example of *movie*, whose synset consists of {*movie, film, picture, moving picture, moving-picture show,* *motion picture, motion-picture show, picture show, pic, flick*}. WordNet also provides a set of *coordinate terms* which are nouns or verbs that have the same hypernym as the target word. Once again *movie* has a number of coordinate terms including *stage dancing*, *attraction*, *performance*, *burlesque*, *play*, and *variety show*.

A final source of data is user tags which are adjectives or verbs, and were not used in the construction of the taxonomy. These tend to be descriptive words that are suitable for use as properties, e.g. *Hungarian*, *fine_tune*, *synchronize*, *semantic*, *amusing*, and *open*. When all of this extra information is added to the taxonomy it results in a greatly enriched ontology which can be used to provide additional services like matching users against one another, or to aid content discovery and recommendation.

As part of the crowdsourcing effort we are planning to enable the export of the generated ontologies and the URLs to which they apply. This would include a list of topics and descriptions where available, as well as relations to other topics of interest. The semantic metadata for each URL would be stored on our servers and made available through an API.

The second form of metadata creation makes use of other vocabularies that have been mapped to WordNet as in the mapping tool MaDaME, which can contribute schema.org and SUMO metadata for any bookmarked URL.

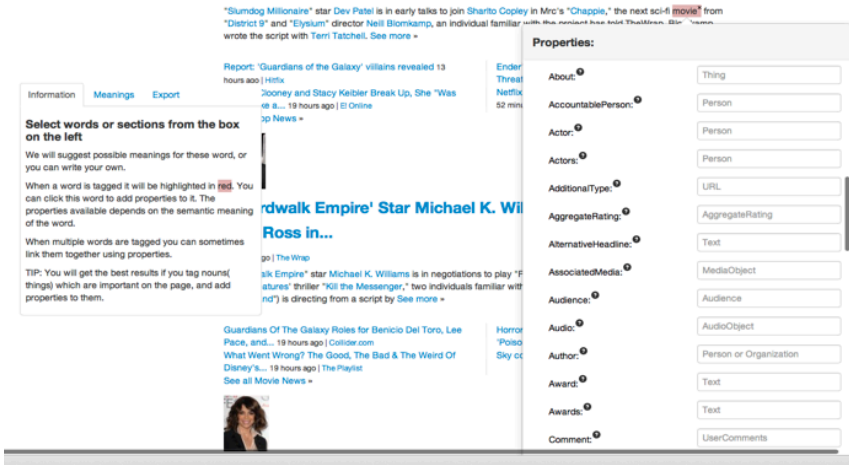
Consider for example one URL bookmarked in the LexiTags data, http://www.imdb.com. This site was tagged with *trailers, information,* and *movies.* When submitted to MaDaME, it can automatically generate markup that can be inserted into the HTML site, or provided as additional data through the aforementioned information API. The mappings generated from the tags are shown in fig. 9. Note that the <span id> is currently assigned to random words at the beginning of the text, and this needs to be inserted into an appropriate location in the original HTML if it is used to mark up the page directly. The example shows that MaDaME currently generates markup from three vocabularies. The original WordNet synset is preserved, as well as the corresponding SUMO class. The SUMO class mappings tend to be quite precise because there is an extensive set of mappings readily available for SUMO[9]. On the other hand the mappings to schema.org are significantly more sparse, because the schema.org types are considerably fewer in number. In cases where no exact match is found a heuristic procedure is used to determine the closest match, and this can result in overly general or erroneous mappings. For example *trailer, preview* is mapped to *schema:Intabgible* whereas a more appropriate mapping might be to the concept *schema:VideoObject* (CreativeWork > MediaObject > VideoObject). On the other hand, many concepts simply do not have a more precise mapping in schema.org, and the mapping of *information, data* to *schema:Intangible* appears to be correct.

[9] http://sigma-01.cim3.net:8080/sigma/Browse.jsp?kb=SUMO&lang=EnglishLanguage

```
<span id="madame-NewsDesk-1" class="tagged"
typeof="sumo:Advertising schema:Intangible wn:synset-preview-
noun-1" about="http://csaba.dyndns.ws:3000/load?
q=51dbf00cc6765c3498000002#madame-NewsDesk-1" data-original-
title="">NewsDesk</span>
<span id="madame-movie-1" class="tagged"
typeof="sumo:MotionPicture schema:Movie wn:synset-movie-
noun-1" about="http://csaba.dyndns.ws:3000/load?
q=51dbf00cc6765c3498000002#madame-movie-1" data-original-
title="">movie</span>
<span id="madame-familiar-1" class="tagged"
typeof="sumo:FactualText schema:Intangible wn:synset-data-
noun-1" about="http://csaba.dyndns.ws:3000/load?
q=51dbf00cc6765c3498000002#madame-familiar-1" data-original-
title="">familiar</span>
```

Fig. 9. Automatic schema.org and SUMO annotation for imdb.com

The key point is that metadata from various different namespaces can automatically be made available for different consumers, simply as a side effect of bookmarking. Search engines can use the schema.org markup, while other services can use the SUMO or WordNet classifications. If the bookmarking service were to grow in popularity, it could become a large repository of schema.org markup for a large set of URLs. Search engines could, presumably, use this markup as if it was embedded in the web sites themselves. But MaDaME also provides a user interface that can be used to greatly enhance the schema.org markup using a drop down form as shown in fig. 10. The form includes a text box for all the possible properties of the schema type which is selected for a word in the text. Users could select words in addition to the tags already assigned, and add these to the schema. The figure also shows the extended markup generated by the tool for the word *movie*.



"Slumdog Millionaire" star Dev Patel is in early talks to join Sharlto Copley in Mrc's "Chappie," the next sci-fi movie" from "District 9" and "Elysium" director Neill Blomkamp, an individual familiar
wrote the script with Terri Tatchell. See more »

Report: 'Guardians of the Galaxy' villains revealed 13 hours ago | Hitfix

Information | Meanings | Export

Select words or sections from the box on the left

We will suggest possible meanings for these word, or you can write your own.

When a word is tagged it will be highlighted in red. You can click this word to add properties to it. The properties available depends on the semantic meaning of the word.

When multiple words are tagged you can sometimes link them together using properties.

TIP: You will get the best results if you tag nouns(things) which are important on the page, and add properties to them.

rdwalk Empire' Star Michael K. Wil
Ross in...

Guardians Of The Galaxy Roles for Benicio Del Toro, Lee Pace, and... 19 hours ago | Collider.com
What Went Wrong? The Good, The Bad & The Weird Of Disney's... 19 hours ago | The Playlist
See all Movie News »

Properties:

About:	Thing
AccountablePerson:	Person
Actor:	Person
Actors:	Person
AdditionalType:	URL
AggregateRating:	AggregateRating
AlternativeHeadline:	Text
AssociatedMedia:	MediaObject
Audience:	Audience
Audio:	AudioObject
Author:	Person or Organization
Award:	Text
Awards:	Text
Comment:	UserComments

```
<span id="madame-movie-1" class="tagged"
typeof="sumo:MotionPicture schema:Movie wn:synset-movie-
noun-1" about="http://csaba.dyndns.ws:3000/load?
q=51dbfcbdc6765c3498000003#madame-movie-1" data-original-
title="">movie<span class="property" property="schema:about"
data-range="Thing" data-comment="The subject matter of the
content." href="Chappie"></span><span class="property"
property="schema:accountablePerson" data-range="Person" data-
comment="Specifies the Person that is legally accountable for
the CreativeWork." href="Sharlto COpley"></span><span
class="property" property="schema:actor" data-range="Person"
data-comment="A cast member of the movie, TV series, season,
or episode, or video." href="Dev Patel"></span><span
class="property" property="schema:comment" data-
range="UserComments" data-comment="Comments, typically from
users, on this CreativeWork." href="Not yet in production"></
span><span class="property" property="schema:director" data-
range="Person" data-comment="The director of the movie, TV
episode, or series." href="Neill Blomkamp"></span></span>
```

Fig 10. Drop down forms and the extensive schema.org markup they can generate

5. Related Work

WordNet is an extremely highly cited resource in all language related areas of study. The official web site at Princeton University maintains a list of publications[10] based

[10] http://wordnet.princeton.edu/wordnet/publications/

on WordNet, but this is no longer maintained because it was “growing faster than it was possible to maintain”.

Within the Semantic Web community WordNet has enjoyed a duality with some researchers criticising its use as an ontology [13]-[15] while others embracing it either as a core taxonomy [16] or as a way to infer semantic relations (e.g. [17], [18]).

[10] used WordNet to automatically infer hierarchical classifications in textually annotated images, and [19] uses it to implement hierarchical faceted classification of recipes and medical journal titles. Both systems use automated extraction and disambiguation of key input terms, which differs from our approach where we ask users to supply these terms. But they use a very similar pruning algorithm to establish the final taxonomic structure.

The idea that free form user tags can be semantically enhanced has received a great deal of attention. Most of the existing work focuses on automatically enriching the tags already present, by exploiting the statistical regularities in the way tags are assigned to resources by users. [20] suggests that the efforts can broadly be classified as (a) extracting semantics of folksonomies by measuring relatedness, clustering, and inferring subsumption relations or (b) semantically enriching folksonomies by linking tags with professional vocabularies and ontologies, for example Wikipedia, and WordNet [21]-[23]. These resources are used in various ways, including to effectively cluster tags, for disambiguation, adding synonyms, and linking to annotated resources and ontology concepts. During this process the terms of the folksonomy are cleaned up and disambiguated, linked to formal definitions and given properties which make them more useful as ontologies.

There are also a few studies in which users are expected to contribute semantics at the time of tagging. [24] studies a corporate blogging platform which included a tagging interface. The tagging interface was linked to a domain ontology, and whenever someone typed a tag that had interpretations in the ontology the interface would present a choice of possible concepts to link the tag to. The ontology would also evolve as users typed new tags which were initially not in the ontology, but the scope of defined tags was limited by the ontology. [25] discuss a sophisticated Firefox plugin, Semdrops, which allowed users to annotate web resources with a complex set of tags including category, property, and attribute tags. These were aggregated in a semantic wiki of the user’s choosing. [26] reports on an open source bookmarking application (SemanticScuttle) that has been enhanced with *structurable* tags, which are tags that users can enhance with inclusion and equivalence relations at the time of tagging. [27] describes extreme tagging in which users can tag other tags, to provide disambiguation and other relational information about tags.

These latter approaches require users to learn new ways of tagging, which are often more complex and opaque than free form tags. The benefit of LexiTagging is that the process is minimally different from activities they are already comfortable with. They simply sign up to a bookmarking site, install a bookmarklet and start tagging. The only addition to the workflow is to disambiguate tags, but this process is so similar to looking up definitions in a dictionary that it needs no explanation.

6 Conclusion

The LexiTags platform is a familiar bookmarking platform, like delicious.com, where users can store the URLs of interesting web sites and tag them with meaningful terms that aid in successive recall and discovery. The only modification is that the user tags are simple dictionary words, not disambiguated strings. But this small change gives the resources on the platform a sound semantic grounding which can significantly enhance the functionality of the service. Some examples of benefits to users are automatic content classification and browsing, external content recommendation, enhanced content discovery, and user profile matching. Some of these services represent crowdsourcing solutions to existing problems which are difficult to fully automate. For example we have shown how the lexitags can be used to infer schema.org and SUMO classifications for each bookmarked web site, which is a task that would otherwise be done manually.

The vision is to create an integrated platform where users begin by simply bookmarking web sites, but then automatically receive the benefits of the enhanced services already described. This gives them the incentive to invest in the added effort to disambiguate their tags. Many of the components are in place, but they need some programming effort to complete the integration. This paper presented the theoretical motivation behind the work, and some preliminary results to show what is possible.

7 References

[1] C. Shirky, “Ontology is Overrated--Categories, Links, and Tags,” *http://www.shirky.com/writings/ontology_overrated.html*, 2007.

[2] A. Mathes, “Folksonomies-cooperative classification and communication through shared metadata,” *Computer Mediated Communication*, 2004.

[3] A. Sheth and K. Thirunarayan, *Semantics-empowered Data, Services, and Sensor and Social Webs*. Morgan & Claypool Publishers, 2012.

[4] H. Hedden, “How SEMANTIC TAGGING Increases Findability,” *EContent magazine*, 08-Oct-2008.

[5] C. Veres, “LexiTags: An Interlingua for the Social Semantic Web,” presented at the Alexandre Passant,Sergio Fernández,John Breslin,Uldis Bojārs, (Eds.) Proceedings of the 4th International Workshop on Social Data on the WebIn conjunction with the International Semantic Web Conference (ISWC2011), Bonn, 2011.

[6] C. Fellbaum, *WordNet: An electronic lexical database*. Cambridge, MA.: MIT Press, 1998.

[7] C. Veres, K. Johansen, and A. Opdahl, “SynsetTagger: A Tool for Generating Ontologies from Semantic Tags,” presented at the Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics., 2013.

[8] G. A. Miller, “WordNet: a lexical database for English,” *Communications of the ACM*, vol. 38, no. 11, Nov. 1995.

[9] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and G. A. Miller, “Introduction to wordnet: an on-line lexical database,” *CSL Report*. International journal of lexicography, 1993.

[10] E. Stoica and M. A. Hearst, *Nearly-automated metadata hierarchy creation*. Association for Computational Linguistics, 2004, pp. 117–120.

[11] P. Mika and T. Potter, “Metadata Statistics for a Large Web Corpus,” *LDOW2012, April 16, 2012, Lyon, France*, 16-Apr-2012. [Online]. Available: http://events.linkeddata.org/ldow2012/papers/ldow2012-inv-paper-1.pdf. [Accessed: 11-Jul-2012].

[12] I. Niles and A. Pease, “Towards a Standard Upper Ontology,” presented at the Proceedings of the international conference on Formal Ontology in Information Systems - FOIS '01, New York, New York, USA, 2001, vol. 2001, pp. 2–9.

[13] A. Gangemi, N. Guarino, C. Masolo, and A. Oltramari, “Restructuring wordnet's top-level,” *AI Magazine*, 2002.

[14] A. Gangemi, N. Guarino, C. Masolo, and A. Oltramari, “Sweetening WORDNET with DOLCE,” *AI Magazine*, vol. 24, no. 3, p. 13, Sep. 2003.

[15] A. Oltramari, A. Gangemi, and E. al, “Restructuring WordNet's top-level: The OntoClean approach,” *LREC2002*, 2002.

[16] F. Suchanek, G. Kasneci, and G. Weikum, “Yago: A Large Ontology from Wikipedia and Wordnet,” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 6, pp. 203–217, 2008.

[17] T. H. Duong, T. N. Ngoc, and G. S. Jo, “A Method for Integration of WordNet-Based Ontologies Using Distance Measures,” *KNOWLEDGE-BASED INTELLIGENT INFORMATION AND ENGINEERING SYSTEMS Lecture Notes in Computer Science, 2008, Volume 5177/2008*, Mar. 2008.

[18] J. Kietz and A. Maedche, “A method for semi-automatic ontology acquisition from a corporate intranet,” *Workshop “Ontologies and text*, 2000.

[19] E. Stoica and M. Hearst, “Demonstration: Using wordnet to build hierarchical facet categories,” presented at the ACM SIGIR Workshop on Faceted Search (August 2006)

[20] F. Limpens, F. Gandon, and M. Buffa, “Linking Folksonomies and Ontologies for Supporting Knowledge Sharing,” *Projet ISICIL :Intégration Sémantique de l“Information par des Communautés d”Intelligence en LigneAppel ANR CONTINT 2008 ANR-08-CORD-011-05*, 01-Aug-2009. [Online]. Available: http://isicil.inria.fr/v2/res/docs/livrables/ISICIL-ANR-EA01-FolksonomiesOntologies-0906.pdf. [Accessed: 15-Aug-2011].

[21] L. Specia, “Integrating folksonomies with the semantic web,” *The semantic web: research and applications*, 2007.

[22] Angeletou, Sofia; Sabou, Marta; Specia, Lucia and Motta, Enrico (2007). Bridging the gap between folksonomies and the semantic web: an experience report. In: The 4th European Semantic Web Conference 2007 (ESWC 2007), 3-7 Jun 2007, Innsbruck, Austria.

[23] C. Van Damme, M. Hepp , K. Siorpaes “Folksontology: An integrated approach for turning folksonomies into ontologies,” *In ESWC workshop. Bridging the Gap between Semantic Web and Web 2.0 (2007)*, 2007.

[24] A. Passant, “Using ontologies to strengthen folksonomies and enrich information retrieval in weblogs,” presented at the Proceedings of International Conference on Weblogs, 2007.

[25] D. Torres, A. Diaz, H. Skaf-Molli, and P. Molli, “Semdrops: A Social Semantic Tagging Approach for Emerging Semantic Data,” *IEEE/WIC/ACM International Conference on Web Intelligence (WI 2011)*, Aug. 2011.

[26] B. Huynh-Kim Bang, E. Dané, and M. Grandbastein, “Merging semantic and participative approaches for organising teachers' documents,” presented at the In J. Luca & E. Weippl (Eds.), Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2008 (pp. 4959-4966). Chesapeake, VA: AACE, Vienna, 2008.

[27] V. Tanasescu and O. Streibel, “Extreme tagging: Emergent semantics through the tagging of tags,” *Proceedings of the First International Workshop on Emergent Semantics and Ontology Evolution, ESOE 2007, co-located with ISWC 2007 + ASWC 2007, Busan, Korea, November 12th, 2007*, 2007.

"Dr. Detective": combining gamification techniques and crowdsourcing to create a gold standard in medical text

Anca Dumitrache[1,3], Lora Aroyo[1], Chris Welty[2], Robert-Jan Sips[3], and Anthony Levas[2]

[1] VU University Amsterdam
anca.dumitrache@student.vu.nl, lora.aroyo@vu.nl
[2] IBM Watson Research Center, New York
cawelty@gmail.com, levas@us.ibm.com
[3] CAS Benelux, IBM Netherlands
robert-jan.sips@nl.ibm.com

Abstract. This paper proposes a design for a gamified crowdsourcing workflow to extract annotation from medical text. Developed in the context of a general crowdsourcing platform, *Dr. Detective* is a game with a purpose that engages medical experts into solving annotation tasks on medical case reports, tailored to capture disagreement between annotators. It incorporates incentives such as learning features, to motivate a continuous involvement of the expert crowd. The game was designed to identify expressions valuable for training NLP tools, and interpret their relation in the context of medical diagnosing. In this way, we can resolve the main problem in gathering ground truth from experts – that the low inter-annotator agreement is typically caused by different interpretations of the text. We report on the results of a pilot study assessing the usefulness of this game. The results show that the quality of the annotations by the expert crowd are comparable to those of an NLP parser. Furthermore, we observed that allowing game users to access each others' answers increases agreement between annotators.

Keywords: crowdsourcing, gold standard, games with a purpose, information extraction, natural language processing

1 Introduction

Modern cognitive systems require human annotated data for training and evaluation, especially when adapting to a new domain. An example of such system is Watson QA [1] developed by IBM, that won the Jeopardy TV quiz show against human competitors. To tune its performance, Watson was trained on a series of databases, taxonomies, and ontologies of publicly available data [2]. Currently, IBM Research aims at adapting the Watson technology for question-answering in the medical domain, which requires large amounts of new training and evaluation data in the form of human annotations of medical text. Two issues arise in this context: (1) the traditional way of ground-truth annotations is slow, expensive and generates only small amounts of data; (2) in order to achieve high inter-annotator agreement, the annotation guidelines are too restrictive. Such practice has proven to create over-generalization and brittleness [3], through losing the sense of diversity in the language, which leads to the fact that natural language processing tools have problems in processing the ambiguity of expressions in text, especially critical in medical text.

The diversity of interpretation of medical text can be seen at many levels; as a simple example, consider the sentence, "Patients exhibiting *acute tailbone pain* should be examined for extra bone nodules." Human experts disagree routinely on whether "acute tailbone pain", "tailbone pain", or "pain" is the primary term in this sentence. Proponents of "tailbone pain" argue that there is a medical term for it (*Coccydynia*) making it primary, others argue that it is pain which is located in the tailbone. Traditional methods of gathering ground truth data for training and evaluation fail to capture such interpretation diversity, leading us to the innovative Crowd Truth approach [4] providing context for this work.

Our analysis led us to believe that the diversity of interpretation occurs at two levels, depending on whether the context is being considered. Term identification, as exemplified in the example above, may be done independent of the clinical context, for example when processing a textbook for background knowledge. However, in the presence of a particular patient, the role of the location and duration modifiers (e.g. tailbone, acute, resp) may or may not be important. We also observe that context-independent tasks tend to require less expertise, allowing us to use a lay crowd more effectively.

These two types of annotation tasks can be performed by two different types of crowds in order to optimize the time, effort and the quality of the final result. Given the experience [4, 5] with defining micro-tasks for the general crowd via crowdsourcing platforms such as Amazon Mechanical Turk[4], or CrowdFlower[5], in this paper we focus on method to engage a crowd of medical experts to be able to resolve *Semantic Ambiguity* in medical text. Annotating complex medical text could be a time consuming and mentally taxing endeavor, therefore the monetary incentive might not be sufficient for attracting a crowd of experts. However, providing a tailored experience for medical professionals through features such as e-learning, and competition with peers, could serve as additional motivation for assembling the right crowd for our task. This can be accomplished by incorporating gamification features into our application.

In this paper, we propose a gamified crowdsourcing application for engaging experts in a knowledge acquisition process that involves domain-specific knowledge extraction in medical texts. The goal of such text annotations is to generate a gold standard for training and evaluation of IBM Watson NLP components in the medical domain. First, we position our work in the context of already existing games with a purpose, crowdsourcing and other niche-sourcing initiatives. Then we outline our approach by focusing on the gaming elements used

[4] www.mturk.com

[5] www.crowdflower.com

as incentives for medical experts, in the context of the overall game application architecture. We show how this gaming platform could fit together with a micro-task platform in a joint workflow combining efforts of both expert and non-expert crowds. Next, we describe the experimental setup to explore the feasibility and the usability of such an application. Finally, we discuss the results of the pilot run of our application, and we identify the points of improvement to bring in future versions.

2 Related Work

In recent years, crowdsourcing has gained a significant amount of exposure as a way for creating solutions for computationally complex problems. By carefully targeting workers with gaming elements and incentives, various crowdsourcing applications were able to garner a significant user base engaged in their tasks. The ESP Game [6] (later renamed Google Image Labeler) pioneered the field by implementing a gamified crowdsourcing approach to generate metadata for images. The reCAPTCHA [7] application combined the CAPTCHA security measure for testing human knowledge with crowdsourcing, in order to perform text extraction from images. The gamified crowdsourcing approach has been employed successfully even in scientific research, with applications such as Galaxy Zoo [8] using crowd knowledge to perform image analysis and extract observations from pictures of galaxies. All of these systems employ mechnisms for a continuous collection of a large amount of human annotated data.

A crowdsourcing framework by [9] introduces 10 design points for Semantic Web populating games. In the context of our research, of a particular interest are: identifying tasks in semantic-content creation, designing game scenarios, designing an attractive interface, identifying reusable bodies of knowledge, and avoiding typical pitfalls. As not all crowdsourcing tasks are suitable for redesign as part of a gamified platform, identifying which of these tasks could engage successfully medical expert crowd is of a key importance to our research. It is also crucial to involve mechanisms to optimize the ratio of time spent and quality and volume of the output [9]. External knowledge sources for annotations (e.g. vocabularies, NLP parsers) can be used to target the work of the players to problems that are too complex to be handled only by computers [9]. Finally, in order to ensure the quality of the answers, unintentional mistakes of the users need to be avoided through clear instructions in the interface [9].

Gamification as applied to text annotation crowdsourcing is an emerging field in different domains. For instance, the Phrase Detective project [10] uses gamified crowdsourcing for building anaphoric annotation ground truth. The input documents are general purpose, and the crowd is not specialized. Two interesting features we considered for Dr. Detective as well, (1) the need for a user training task to improve the usage of the application, and (2) understanding of the user profile (e.g. players can examine a considerable variation in their interaction styles, abilities or background knowledge.

The Sentiment Quiz [11], played through various social networking platforms, employs crowdsourcing to evaluate accuracy of sentiment detecting algorithms over sentences, and to create a lexicon of sentiments in various languages. The requirements for user incentives in *Dr. Detective* were based on the analysis provided by Sentiment Quiz, e.g. for scoring, high score board, and level-based goals, as well as for enhancing the crowd output through statistical methods applied in the disagreement analytics.

However, neither the Sentiment Quiz, nor the Phrase Detective applications actively seek out to capture the ambiguity in language. Phrase Detective even tries to enforce agreement, by awarding additional points for annotators that agree with the ground truth. Neither do most applications in the domain study the effect of using specialized crowds to perform the information extraction tasks. Our goal is to build an end-to-end gamified crowdsourcing platform that can capture disagreement between annotators, while catering specifically to experts in the medical field.

3 "Crowd-Watson" Architecture: The Game Perspective

In this section, we describe the architecture for *Dr. Detective* [6] – an application for engaging experts in knowledge extraction tasks for creating ground truth annotations in medical texts. We start by framing *Dr. Detective* as part of the general *Crowd-Watson*[7] framework for crowdsourcing medical text annotation [12]. Then, we tackle the challenge of tailoring the application to a specialized crowd of medical professionals, through a study of possible motivating factors. Finally, we describe how gamification elements were integrated with the crowdsourcing workflow.

The *Crowd-Watson* framework supports the composition of crowd-truth gathering workflows, where a sequence of micro-annotation-tasks can be executed jointly either by the general crowd on platforms like CrowdFlower, or by specialized crowd of domain experts on gaming platform as Dr. Detective. Crowd-Watson framework focuses on micro-tasks for knowledge extraction in medical text. The main steps involved in the *Crowd-Watson* workflow are: **pre-processing** of the input, **data collection**, **disagreement analytics** for the results, and finally **post-processing**. These steps are realized as an automatic end-to-end workflow, that can support a continuous collection of high quality gold standard data with feedback loop to all steps of the process. The input consists of medical documents, from various sources such as Wikipedia articles or patient case reports. The output generated through this framework is annotation for medical text, in the form of concepts and the relations between them, together with a collection of visual analytics to explore these results. The architecture of this application, and the way its components interact with each other, can be seen in Figure 1. In this paper, we focus on those aspects of the architecture that relate to the *Dr. Detective* gaming platform for data collection. A full description of the *Crowd-Watson* architecture is available at [12].

[6] `http://crowd-watson.nl/dr-detective-game`

[7] `http://crowd-watson.nl`

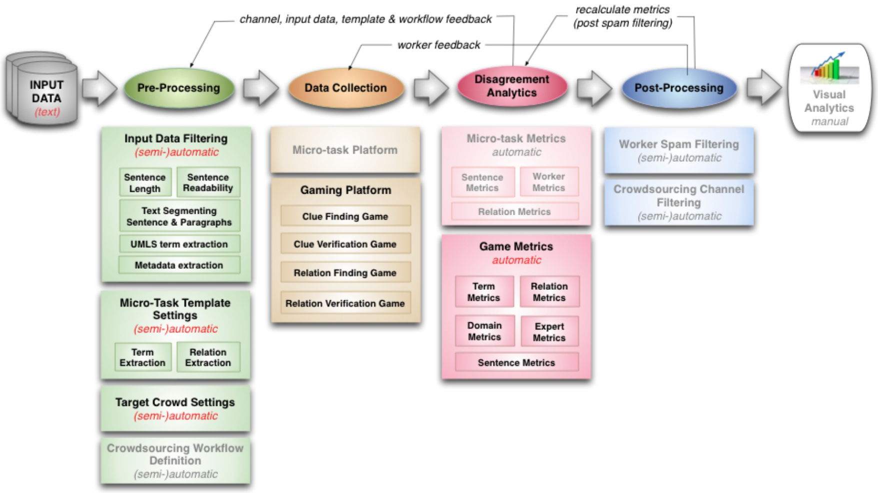


Fig. 1. Crowd-Watson Framework Design *(the highlighted components are the ones related to the Game Platfom)*

3.1 Pre-Processing for the Game Platform

Typically, the input is available in an unstructured format (e.g. simple text). As part of the **input data filtering** step, additional metadata, such as the specialization field in which it was published or, for case reports, the diagnosis of the patient, can be extracted from these documents. In addition, some annotation can also be generated automatically, by mapping the text to the UMLS vocabulary of biomedical terminology, classification, and coding standards [13]. The UMLS parser can be used to identify both concepts and relations, however, as a fully automated approach, it suffers from the typical issues of NLP techniques [14], such as lack of contextual awareness, and limited ambiguity processing capabilities. Nevertheless, UMLS annotations can be employed as a good baseline for measuring the efficiency of the crowdsourced answers.

The workers are asked to perform a series of annotation tasks on the input documents. The purpose of these tasks is creating annotation in the form of concepts and the relations between them. We define these tasks according to four **micro-task templates**:

1. *Term extraction* – the task of identifying all the relevant terms in a text, where a term refers to a set of words that forms a coherent medical concept;
2. *Term categorization* – the task of classifying a medical term into an appropriate category, such as the concepts in the UMLS thesaurus;
3. *Relation extraction* – the task of identifying whether or not a relation exists between two medical terms;

4. *Relation categorization* – the task of classifying a medical relation into an appropriate category (or set of categories), such as the relations in the UMLS thesaurus.

The workers on *Crowd-Watson* consist of both an expert crowd, and a general crowd. Each of these crowds interacts with the input documents on a specialized platform – for the general crowd, regular crowdsourcing micro-tasks have been constructed on CrowdFlower, whereas the expert crowd employs the *Dr. Detective* application for solving tasks tailored to their profile. The tasks can be solved by both the general, and the expert crowd. The **target crowd setting** step entails picking the difficulty level of the task according to the level of expertise of the crowd. For instance, when discussing term extraction, the general crowd could reliably find demographic terms, as they do not require significant medical knowledge, whereas the expert crowd can focus on annotating more difficult terminology.

3.2 Game Disagreement Analytics

After the input data is formated and filtered appropriately through the *preprocessing* components, it is sent to the *data collection* component to to gather either expert annotation (through the gaming platform) or lay crowd annotations (through the micro-task platform). Next, the annotation results are analyzed with a set of content and behavior-based metrics, to understand how the disagreement is represented in both cases [15, 16], and to assess the quality of the individual workers, and the quality of the individual and overall crowd truth results.

To track the individual performance of a user in the crowd, the expert metrics were developed. For each sentence in the input, the performance of the worker can be measured as a set of vectors, according to the task they solved on that input. Such a vector is composed of 0 and 1 values, such that for each answer a user annotated in that sentence, there is a 1 in the corresponding position, whereas answers that were not picked by the user are set to 0. These answer vectors can also be measured at the level of the domain.

At the level of the sentence, a set of task-dependent sentence metrics were also defined. For either term extraction or relation extraction, any sentence can be expressed as a sentence vector – the sum of all the individual user vectors on that sentence, for that task. Furthermore, an added layer of granularity can be introduced by considering the categories for the terms and relations. This representation can then be used to define appropriate metrics for sentence clarity, what the popular answers were, how disagreement is represented, and similarity of annotation categories and domains.

The prime role of the disagreement analytics in the gaming platform are to provide explicit measures for the quality and completeness of the final result; to identify gaps of missing types of annotations; or to discover possible contradictions and inconsistencies. This is opposed to the micro-task disagreement analytics, which follow the same approach but apply to filters for spam identification.

4 Data Collection: Gaming Platform

In order to collect data from a crowd for medical experts, it is imperative to find the necessary motivators for engaging them into contributing. To this end, we have performed a series of qualitative interviews with medical students and professionals. The purpose was to identify what requirements and features would the medical crowd be interested in seeing in a crowdsourced application, and how this application could be built to help in their work. These interviews established incentives for crowd labor [17], such as competition, learning, and entertainment in the context of working in the medical field, as well as documents that the medical crowd would be interested in reading.

After discussing with 11 people in the medical field (2 professionals, 3 lecturers, 5 students), we were able to identify several key requirements to incorporate into the gaming platform:

- at the level of the input, the interviewees expressed their interest in **reading medical case reports**;
- **learning** about their field, through targeted micro-tasks and extended feedback on their answers, was the most significant motivator;
- the interviewees expected the tasks to challenge their **problem-solving** skills;
- **competition** with peers emerged as a secondary motivator;
- the tasks need to be fun to solve, making **entertainment** as another secondary motivator;
- medical professionals have difficult schedules, and would prefer to have **flexibility** in the time required to engage with the application;

In order to attract users to the application, a goal that is seen as useful by the players needs to be firmly established. As *learning* proved to be the most relevant incentive from the interviews, we focused the goal of the application on this, while also trying to incorporate the *problem-solving* requirement. We developed the concept of a **clue-finding game**, where the text annotation tasks were put in the context of searching for clues in the history of a patient. For instance, when performing the task of term extraction on a patient case report, the user can annotate any of these three **clue types**:

1. the term is a clue *leading* to the final diagnosis of the case;
2. the term is a false clue that is *irrelevant* to the final diagnosis of the case;
3. the term is a *normal condition* that does not influence the final diagnosis of the case.

The clue types can be used as an incentive, involving users with the task they are solving by redesigning it as a medical puzzle, but it can also be used to generate additional annotation. The annotations retrieved from the general crowdsourcing approach are dependent on the context of the sentence where they were identified, so by asking the expert crowd to find meta-relations at the level of the document, we can generate knowledge that is valid generally for the domain. This kind of task cannot be solved simply with the use of contextual information, and requires background knowledge of the field, therefore making it suitable for an application targeted at experts.

The qualitative interviews helped us identify the extrinsic motivators for engaging the medical crowd. After the goal of the application was established, the final step was translating the user incentives into concrete features for building the *Dr. Detective* gaming platform.

4.1 Difficulty

In order to support the user learning experience and introduce flexibility in task solving, we define the concept of difficulty. This refers to the combination of skill and time required for reading the document, and then performing the annotation task. While it is difficult to hypothesize on the comparative difficulty of performing annotations, the difficulty of the document can expressed as syntactic and semantic difficulty. The syntactic difficulty expresses the effort need for reading the document in three components: the *number of sentences* in the document (NoS), the *number of words* (NoW), and the *average sentence length* (ASL). The semantic difficulty expresses the effort needed for understanding the text in two components: the *number of UMLS concepts* present in the document ($NoUMLS$), and the *readability* of the document ($SMOG$). The SMOG [18] formula for computing readability was employed, as it is often recommended for use in evaluating healthcare documents [19]. Therefore, for every document D, its difficulty is defined as the norm of the normalized five-component vector:

$$difficulty(D) = \|(NoS, NoW, ASL, NoUMLS, SMOG)\|.$$

4.2 Scoring

In order to develop the *competition* incentive, a scoring system was devised, to reward players for their work. Through viewing a high score board, they are also encouraged to compete against each other.

We want to reward users when they perform in a way that is beneficial to us. We want to collect the correct answers to the task, therefore, selecting a high-consensus solution should yield more points. This strategy could, however, make users rely entirely on the answers of others. Therefore, in order to encourage a wider answer set and capture semantic ambiguity, we need to give points for newly discovered answers. Users should also be penalized for giving wrong answers. We also want to encourage users to return to the application, and keep playing. Finally, in order for users to solve tasks in increasing difficulty, scoring needs to be proportional to the difficulty for solving the task [20]. Based on this, for each user U solving a task T on document D, we developed the following scoring components:

- *popular*(U, D, T): the points users receive if they make annotations that were previously selected by at least one other user; we also want to reward partial answers, in order to capture ambiguity;
- *consecutive*(U): the points users gain the more consecutive tasks they solve;

- $disovered(U, D, T)$: the points users receive if they are the first to discover an answer, if it is then selected by at least one other user;
- $wrong(U, D, T)$: the points users lose if their answers are not selected by any other user.

Based on this analysis, we developed the following scoring formula:

$$\begin{aligned} score(U, D, T) = difficulty(D)\cdot \\ \cdot (popular(U, D, T) + consecutive(U) \\ + discovered(U, D, T) - wrong(U, D, T)). \end{aligned}$$

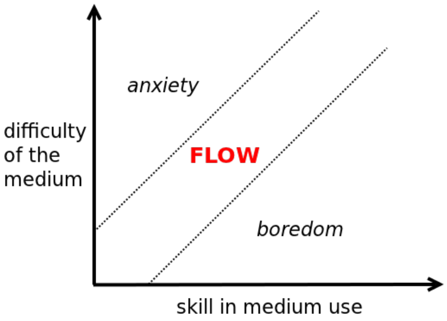


Fig. 2. Game flow as an expression of skill and difficulty

4.3 Immersion

In order to develop the *entertainment* incentive, the crowdsourcing application needs to provide immersion inside the task-solving experience. Immersion is based on the concept of game flow [21], which states that at every point in the game, the difficulty needs to be proportionate with the skill required to solve the task. Skill at playing is acquired by the user as they solve more tasks. If the difficulty is disproportionately large compared to the skill, it will cause anxiety for the user, whereas if the difficulty is too small, the user will be bored. Immersion is achieved when skill and difficulty are proportionally balanced, as illustrated in Figure 2.

Immersion is considered when choosing the next document that the user will be asked to solve as part of the game. When a user solves a task on D_i, the document they will be asked to solve next needs to have a higher difficulty in order to avoid boredom, but the increase needs to be low enough to avoid anxiety. Therefore, we define the set of possible documents that occur after D_i as:

$$\begin{aligned} next(D_i) = \{D_j \mid difficulty(D_j) = min(difficulty(D_i) - difficulty(D_t), \\ \forall t \neq i \text{ where } difficulty(D_t) \geq difficulty(D_i))\} \end{aligned}$$

4.4 Levels

Finally, in order to satisfy the constraint for *flexibility*, game levels were implemented to quantify the skill required for solving the tasks. As skill is proportional with difficulty, we defined the game levels by quantifying the difficulty metric previously described into three intervals:

1. easy: $\{D \mid difficulty(D) \in [0, 2]\}$,
2. normal: $\{D \mid difficulty(D) \in [3, 4]\}$,
3. hard: $\{D \mid difficulty(D) \in [5, 6]\}$.

These levels should enable users to plan which task they want to solve in accordance to the time they have at their disposal, while also providing a goal-based incentive of progressing in their skill [20].

5 Experimental Setup

In order to test the feasability of the *Dr. Detective* setup, we implemented a version of the workflow described in Section 3, and set up a pilot run involving a crowd of medical professionals. As part of our pilot run, we performed an initial evaluation of both the quality of the answers, and the user enjoyment as part of this gamified crowdsourcing platform. The goal of this experiment can be described as three questions, which will be discussed as part of our results:

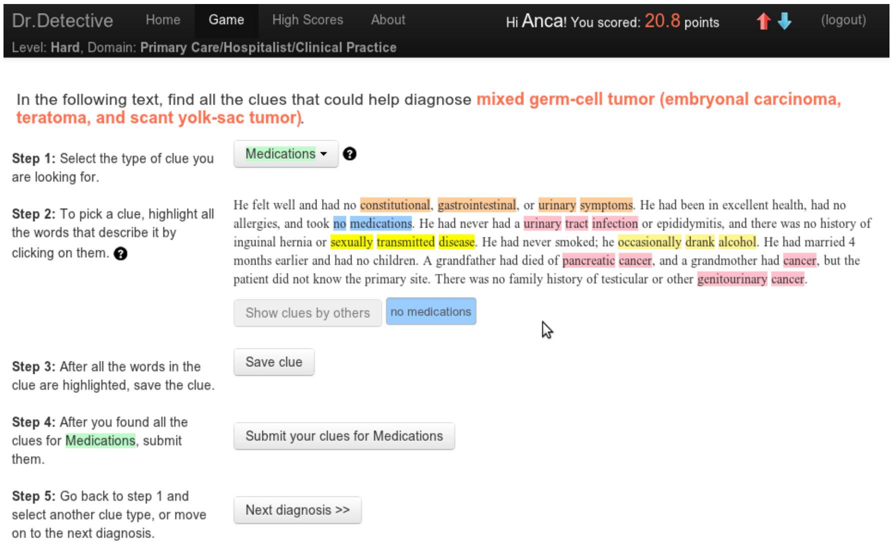
1. How do the answers annotated by the crowd compare to those found by the UMLS parser?
2. Does having access to the answers of other users stimulate diversity of opinion?
3. Did users experience immersion in the gaming experience?

In order to answer these questions, we set up two versions of the game, one in which users had the ability to see the answers of others, and one in which they did not. In addition, some of the gaming elements that would ensure the users keep in the state of game flow (high scores board, next document selection mechanism, levels) were only limited to the full version of the game. We constructed an experiment where the users would play both versions of the game, then answer a questionnaire on their experiences. The details of this experimental setup are described in this section.

5.1 Input

Based on a suggestion in the qualitative interviews, the input was selected from clinical cases published in the New England Journal of Medicine[8]. 10 documents were picked out of four of the most popular specialties (Hematology/Oncology, Nephrology, Primary Care/Hospitalist/Clinical Practice, Viral Infections). The diagnosis was extracted from each document, based on a string matching procedure performed on the text marked in “diagnosis” section headings (e.g. clinical diagnosis, pathological diagnosis etc.). The documents were split into paragraphs, to increase the ease of reading, and the difficulty metrics (described in

[8] `www.nejm.org`



Level: **Hard**, Domain: **Primary Care/Hospitalist/Clinical Practice**

In the following text, find all the clues that could help diagnose **mixed germ-cell tumor (embryonal carcinoma, teratoma, and scant yolk-sac tumor)**.

Step 1: Select the type of clue you are looking for.

Medications ▾ ?

Step 2: To pick a clue, highlight all the words that describe it by clicking on them. ?

He felt well and had no constitutional, gastrointestinal, or urinary symptoms. He had been in excellent health, had no allergies, and took no medications. He had never had a urinary tract infection or epididymitis, and there was no history of inguinal hernia or sexually transmitted disease. He had never smoked; he occasionally drank alcohol. He had married 4 months earlier and had no children. A grandfather had died of pancreatic cancer, and a grandmother had cancer, but the patient did not know the primary site. There was no family history of testicular or other genitourinary cancer.

Show clues by others | no medications

Step 3: After all the words in the clue are highlighted, save the clue.

Save clue

Step 4: After you found all the clues for Medications, submit them.

Submit your clues for Medications

Step 5: Go back to step 1 and select another clue type, or move on to the next diagnosis.

Next diagnosis >>

Fig. 3. Screenshot from the "Dr. Detective" game

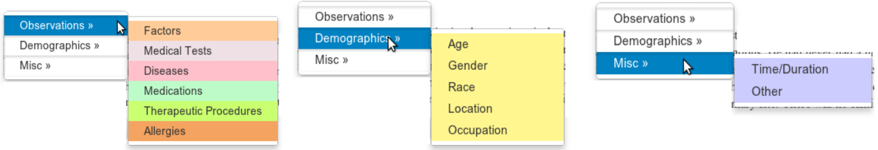


Fig. 4. Term types in the game.

Section 4.1) were then applied to each paragraph. Finally, we selected a set of 20 paragraphs, with the values in the difficulty vector uniformly distributed to represent a broad range of text types, to use for the game, as we wanted to ensure that all of the text would be annotated in the limited time frame of the experiment run.

5.2 Task

The micro-task templates (described in Section 3.1) selected for this pilot were (1) term extraction, and (2) term categorization. Based on how relevant they are at describing patient case reports, 3 meta-types, each with a set of term types taken from UMLS, were selected and implemented in the interface for the categorization task. These term types are based on factor categories given to domain experts during the expert annotation phase for Watson. The type selection menu can be seen in Figure 4. In total, 13 term types were available for the users to annotate. As most interviewers expressed their interest in a problem-solving application, we decided to set the clue type user seek as part of the application (described in Section 4) to (1) the term is a clue *leading* to the final diagnosis of the case. Finally, in order to encourage the diversity of opinion, and therefore capture ambiguity, we allowed users to look at the answers of others for the task they are solving. This feature was made available through a button, which the users could choose to press in order to toggle the other answers. The scoring formula (described in Section 4.2) ensures that users are motivated to find new answers even in this circumstances, through the use of discovery bonus points. The users could access the details of how their score was computed through a hover notification in the menu. An example of how this task was presented to the users as part of the *Dr. Detective* interface can be seen in Figure 3.

5.3 Users

The pilot run of the *Dr. Detective* game had 11 participants in total, with 10 players engaging with the full game version, and 7 engaging with the simple version. In total, 155 annotation sets were collected, with each paragraph solved as part of 2 to 7 different game rounds. In addition, 6 players completed the feedback questionnaire.

6 Results and Discussion

In keeping with the research questions defined in the previous section, we first analyzed how the answers from the crowd compare to the results of the UMLS parser. We selected the top three paragraphs that were played the most, and compared the answers to the term list generated by the UMLS MetaMap parser [9] for the same paragraphs. Fig. 5 shows the crowd was able to identify the majority of the words annotated with UMLS. Additionally, Fig. 6 shows that around one third of the terms in UMLS had a full match with terms annotated by the crowd. Factoring in the partial term matches, the crowd was able to identify most of the

[9] http://metamap.nlm.nih.gov/

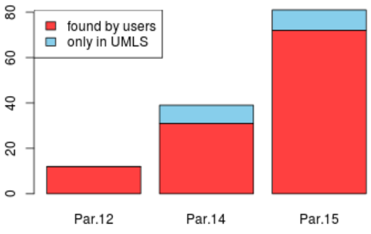


Fig. 5. Words in UMLS for the 3 most popular paragraphs in the game

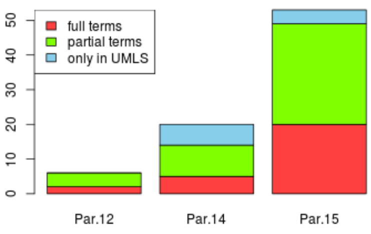


Fig. 6. Terms in UMLS for the 3 most popular paragraphs in the game

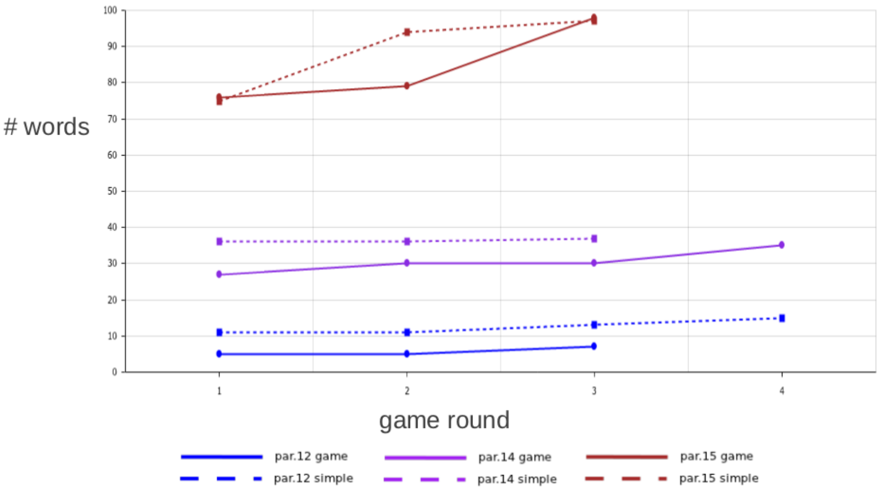


Fig. 7. Number of words for the 3 most popular paragraphs, after each round of each game version

UMLS terms. This shows the efficiency of the crowd answers is quite high, enough for the crowd to be considered as a viable alternative to automated named-entity recognition, provided that enough users give their input for a paragraph.

Next, we look at how diversity of opinion was expressed by the game users. Specifically, we are interested in finding out whether being able to see the results of other people will stimulate disagreement, or rather make users select each other's answers. In other to achieve this, we look at how the answers per paragraph varied according to the version of the game that the user played.

Fig. 7 shows how the number of new words per paragraph increases after each round of the game, for the top three paragraphs. Each version of the game seems to follow the same progression in the rate of new words identified, with the first users finding most of the words, and then only slight increases as the paragraph is played by other people. However, the simple version of the game seems to constantly feature a higher total word count, as opposed to the full game version. The same trend was observed both for the number of new types, and the number of distinct terms. This seems to indicate that the full game version was less encouraging for collecting a wide array of terms.

In order to rule out an issue related to some other feature in the full game version, we looked at how the behavior of pressing the button to view other answers affected the output. Out of 67 game rounds played in the full version, this button was only pressed in 18 of the rounds, so it appears this was not a popular feature to begin with. Fig. 8 shows that, actually, users tended to annotate more words in total when they pressed. However, as evidenced in Fig. 9, the ratio of new words to total words in this case was much lower than when the button was not pressed. Additionally, it appears there is not much difference

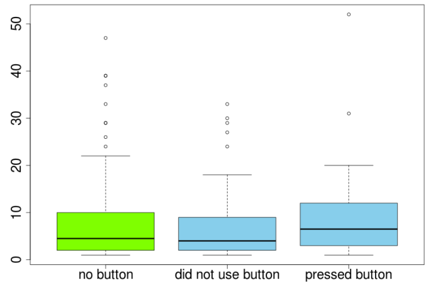


Fig. 8. Ratio of total words per round, grouped by the use of the button to view the answers of others

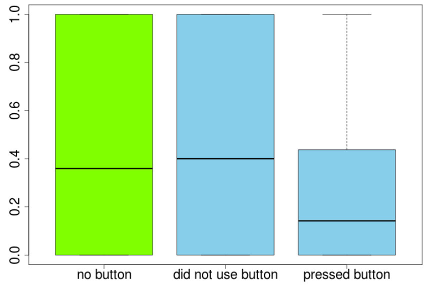


Fig. 9. Ratio of new to total words, grouped by the use of the button to view the answers of others

between the simple version of the game, and the full version, but where the users chose not to look at the answers of others. Therefore we can infer that having access to all the answers makes the crowd act more conservative, selecting less new words, but rather choosing to validate the answers of others.

When looking at the answers in the questionnaire related to the usefulness of seeing other people's annotations, we found that most people (67%) were ambivalent to having the option of checking their answers. Some users reported using this feature as a tool for better understanding the task, while others claimed it validated the answers they had already chosen. Overall, it seems that having access to all the other answers makes users less likely to find and annotate new words, which could mean a loss in the ambiguity of the annotation. It also provides an unfair advantage to the first users to annotate a paragraph, as their score would likely keep increasing as other people keep selecting their answers.

Finally, we analyzed whether immersion in the game occurred for the users involved, and how each individual game feature was rated. The flow of the game was reported to be good, with 83% of the users saying they were neither too bored, or overwhelmed. Most users found the levels to be a useful addition, with 50% being satisfied with the level progression, and 33% being ambivalent to it. However, some users pointed out that they expected more challenge from the advanced level. As the difficulty is currently computed only based on textual metrics, the game could potentially get boring for users. For this reason, domain difficulty should be incorporated in future versions of the game. The scoring part of the game was less well received, with 83% of the users declaring they found the way their score is computed only somewhat clear. Therefore, in future game versions, a more detailed scoring breakdown should be implemented, with users being able to access the history of the cases they solved. Finally, most users reported to have enjoyed the game, and expressed an interest in returning to play, provided they can solve more difficult cases and get more feedback. The full game version was almost universally preferred by the users.

7 Conclusion and Future Work

This paper proposes a design for *Dr. Detective* – a gamified crowdsourcing platform to extract annotation from medical text. *Dr. Detective* was developed in the context of *Crowd-Watson*, a general crowdsourcing framework for extracting text annotation by engaging both a general crowd, and a domain expert crowd. The gaming platform was designed taking into account the requirements of the expert crowd, and illustrating their implementation in a clue finding game. Specific gamification elements were incorporated, such as difficulty, scoring, immersion, and levels. A first version of *Dr. Detective* was implemented and tested. The pilot run showed that the quality of the results of the crowd are comparable to those of an NLP parser. Allowing users to see the answers of others resulted in increased agreement, and thus decreased the desired diversity in answers. The overall user feedback for the application was positive. However, it was clear that users desire more complex challenges in order to keep them engaged.

An important next step is to define and test disagreement metrics that are specific to the gaming environment. As we have seen in previous research, a promising starting point are the disagreement metrics developed for the data collected through the micro-task platform. We also plan to further test how each of the gaming features performs individually, in order to fine-tune the application to understand better their influence on the quality and volume of the end result, as well as to adapt best to the needs of the users. Finally, we will explore how to further integrate the gaming and the micro-task crowdsourcing workflows, by using the output from one workflow to enhance the input for the other (e.g. ask one crowd to perform the term extraction, and the other crowd the relation extraction), or by asking one crowd to validate the output of the other crowd.

Acknowledgements

The authors would like to thank Kathrin Dentler and Dr. Petra Wolffs for their help with finding participants for both the interviews and the application pilot run, as well as the students and medical professionals who were involved in these activities, and who provided their feedback.

References

1. Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A.A., Lally, A., Murdock, J.W., Nyberg, E., Prager, J., Schlaefer, N., Welty, C.: Building Watson: An Overview of the DeepQA Project. AI Magazine **31** (2010) 59–79
2. Kalyanpur, A., Boguraev, B., Patwardhan, S., Murdock, J.W., Lally, A., Welty, C., Prager, J.M., Coppola, B., Fokoue-Nkoutche, A., Zhang, L., et al.: Structured data and inference in DeepQA. IBM Journal of Research and Development **56**(3.4) (2012) 10–1
3. Aroyo, L., Welty, C.: Harnessing disagreement for event semantics. Detection, Representation, and Exploitation of Events in the Semantic Web 31
4. Aroyo, L., Welty, C.: Crowd Truth: Harnessing disagreement in crowdsourcing a relation extraction gold standard. WebSci2013. ACM (2013)

5. Inel, O., Aroyo, L., Welty, C., Sips, R.J.: Exploiting Crowdsourcing Disagreement with Various Domain-Independent Quality Measures. Technical report, VU University Amsterdam (July 2013). `http://crowd-watson.nl/tech-reports/20130702.pdf`
6. Von Ahn, L., Dabbish, L.: Labeling images with a computer game. In: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM (2004) 319–326
7. Von Ahn, L., Maurer, B., McMillen, C., Abraham, D., Blum, M.: recaptcha: Human-based character recognition via web security measures. Science **321**(5895) (2008) 1465–1468
8. Lintott, C.J., Schawinski, K., Slosar, A., Land, K., Bamford, S., Thomas, D., Raddick, M.J., Nichol, R.C., Szalay, A., Andreescu, D., et al.: Galaxy zoo: morphologies derived from visual inspection of galaxies from the sloan digital sky survey. Monthly Notices of the Royal Astronomical Society **389**(3) (2008) 1179–1189
9. Siorpaes, K., Hepp, M.: Games with a Purpose for the Semantic Web. Intelligent Systems, IEEE **23**(3) (2008) 50–60
10. Chamberlain, J., Poesio, M., Kruschwitz, U.: Phrase Detectives - A Web-based Collaborative Annotation Game. In: Proceedings of I-Semantics. (2008)
11. Scharl, A., Sabou, M., Gindl, S., Rafelsberger, W., Weichselbraun, A.: Leveraging the Wisdom of the Crowds for the Acquisition of Multilingual Language Resources. In: Proc. 8th LREC - International Conference on Language Resources and Evaluation. (2012)
12. Lin, H.: Crowd Watson: Crowdsourced Text Annotations. Technical report, VU University Amsterdam (July 2013). `http://crowd-watson.nl/tech-reports/20130704.pdf`
13. Bodenreider, O.: The unified medical language system (UMLS): integrating biomedical terminology. Nucleic acids research **32**(suppl 1) (2004) D267–D270
14. Goldberg, H.S., Hsu, C., Law, V., Safran, C.: Validation of clinical problems using a UMLS-based semantic parser. In: Proceedings of the AMIA Symposium, American Medical Informatics Association (1998) 805
15. Aroyo, L., Welty, C.: Measuring crowd truth for medical relation extraction. In: AAAI 2013 Fall Symposium on Semantics for Big Data (in print). (2013)
16. Soberón, G., Aroyo, L., Welty, C., Inel, O., Lin, H., Overmeen, M.: Crowd Truth Metrics. Technical report, VU University Amsterdam (July 2013). `http://crowd-watson.nl/tech-reports/20130703.pdf`
17. Tokarchuk, O., Cuel, R., Zamarian, M.: Analyzing crowd labor and designing incentives for humans in the loop. IEEE Internet Computing **16**(5) (2012) 0045–51
18. McLaughlin, G.H.: SMOG grading: A new readability formula. Journal of reading **12**(8) (1969) 639–646
19. Doak, C.C., Doak, L.G., Root, J.H.: Teaching patients with low literacy skills. AJN The American Journal of Nursing **96**(12) (1996) 16M
20. Von Ahn, L., Dabbish, L.: Designing games with a purpose. Communications of the ACM **51**(8) (2008) 58–67
21. Sherry, J.L.: Flow and media enjoyment. Communication Theory **14**(4) (2004) 328–347

SLUA: Towards Semantic Linking of Users with Actions in Crowdsourcing

Umair ul Hassan, Sean O'Riain, Edward Curry

Digital Enterprise Research Institute
National University of Ireland,
Galway, Ireland.

{umair.ul.hassan, sean.oriain, ed.curry}@deri.org

Abstract. Recent advances in web technologies allow people to help solve complex problems by performing online tasks in return for money, learning, or fun. At present, human contribution is limited to the tasks defined on individual crowdsourcing platforms. Furthermore, there is a lack of tools and technologies that support matching of tasks with appropriate users, across multiple systems. A more explicit capture of the semantics of crowdsourcing tasks could enable the design and development of matchmaking services between users and tasks. The paper presents the SLUA ontology that aims to model users and tasks in crowdsourcing systems in terms of the relevant actions, capabilities, and rewards. This model describes different types of human tasks that help in solving complex problems using crowds. The paper provides examples of describing users and tasks in some real world systems, with SLUA ontology.

Keywords: crowdsourcing, human computation, users, tasks, ontology

1 Introduction

Collective intelligence systems [1] have demonstrated the use of networked humans and computers for solving complex problem, by applying techniques such as crowdsourcing [2], social computing [3] and human computation [4]. Online marketplaces like Amazon Mechanical Turk[1] provide access to large pool of human workers willing to perform variety of micro-tasks for money. Whilst other platforms focus on domain specific crowd services e.g. uTest[2] provides software testing services.

Most of the existing crowdsourcing platforms are isolated in terms of their users and tasks. People contribute towards either a few popular platforms or the systems relevant to their specific domain of knowledge. Hence human resources may be underutilized due to a lack of tools that help people in finding tasks across multiple crowdsourcing platforms. Similarly, task requesters are unable to query across multi-

[1] http://www.mturk.com

[2] http://www.utest.com

ple platforms for their tasks to find appropriate workers with required skills or knowledge.

Main objective of the SLUA (Semantically Linked Users and Actions) ontology is to define a lightweight model for describing crowdsourcing tasks and users with regard to human capabilities, actions and rewards. The scope of the ontology is limited to the micro-tasks that can be performed with in minutes. The specific aims of SLUA ontology are

- To enable interoperability and reuse among crowdsourcing platforms across the web. For example, an active user on Quora[3] for the topic on Cloud Computing might be the right person to edit a Wikipedia article on the same topic.
- To support people in finding online tasks according to their capabilities and motivation. For example, if a person is knowledgeable about the city of New York, then she can help fix problems in Wikipedia[4] articles or tag images of buildings in New York in Amazon Mechanical Turk.
- To facilitate algorithmic matching of tasks and users according to human capabilities, actions, and rewards. For example, a human computation platform might need to verify the chemical formula of a drug from a chemist with the relevant education listed on LinkedIn[5].

The main contributions of this paper are the initial description of the SLUA ontology and its mappings to other existing ontologies, and examples of various tasks described using SLUA. The rest of this paper is organized as follows. Section 2 motivates the need of ontology for describing users and tasks in crowdsourcing platforms. Section 3 highlights the requirements of ontology according to relevant concepts found in literature. Section 4 provides the description of classes and properties in SLUA ontology. Section 5 details some example usage of SLUA for semantic description of tasks and users. Section 6 discusses related work and Section 6 concludes the paper.

2 Motivation

Crowdsourcing platforms differ from each other in terms of the tasks that humans can perform and the characteristics of human contributors [5]. Wikipedia requires the crowd to create or edit articles by contributing textual content and references. Quora is powered by questions and answers contributed by online users. Both Wikipedia and Quora relay on the fact that people are motivated to contribute to the crowdsourcing efforts because of social good or self-serving motivations. Amazon Mechanical Turk serves as the market place of human services for performing small online tasks in exchange for money. TaskRabbit[6] allows people to outsource their small physical

[3] http://www.quora.com

[4] http://www.wikipedia.org

[5] http://www.linkedin.com

[6] http://www.taskrabbit.com

Table 1. Common terminology used for the concepts of tasks and users in the documentation of popular online marketplaces for crowdsourcing.

Concept	**MTurk**	**Mobileworks**	**Shorttask**	**CrowdFlower**
Task	HIT	Task	ShortTask	Microtask
User	Worker Requester	Worker Developer	Solver Seeker	Contributor Customer
Reward	Payment	Payment	Reward	Payment
Capability	Qualification	Filter		

tasks to other people against small monetary price. Microtask[7] uses online gamers to solve problems typically not solvable by computers. In short, the heterogeneity of crowdsourcing systems exists at task, user, and platform levels. Some tasks require cognitive skills while other need physical abilities from humans. Some tasks reward in terms of money while others compensate through enjoyment.

The heterogeneity of crowdsourcing platforms limits interoperability of applications that access more than one crowdsourcing platform. Furthermore, development of cross platform services becomes difficult due to variations of data semantics for each platform. For instance, there is a lack of search engines for microtasks and existing general search engines fail to address this problem. Similarly existing crowdsourcing platforms do not support any application interfaces for users search based on human capabilities. Recently there has been effort to describe crowdsourcing platforms with the help of taxonomies [6]. However they do not cover the modeling of human tasks, actions, and capabilities whilst describing concepts associated with the design aspects of crowdsourcing platform. Therefore we observe that there is a need for a common language for describing human tasks, actions, rewards, and capabilities in crowdsourcing platforms, as well as their relationships. An appropriate ontology may serve the purpose therefore facilitating interoperability supporting broad range of computation services. In the next section we summarize the conceptual requirements of such ontology and assess the coverage of requirements by existing ontologies.

3 Ontology Requirements

In this section we analyze the requirements of ontology for human tasks in crowdsourcing. The requirements are based on the common terminology found in current crowdsourcing platforms. Table 1 shows a variety of terminology and concepts used among major crowdsourcing marketplaces. This heterogeneity of terminology creates a gap in terms of common understanding of crowdsourcing concepts among users and developers [7]. Additionally, heterogeneity is reflected in the application programming interfaces offered by crowdsourcing platform, resulting in interoperability issues in terms of the semantics of data structures and algorithms [7].

[7] http://www.microtask.com

The limitations due to the heterogeneity of crowdsourcing platforms necessitate development of domain ontology. In this work we focus on defining lightweight domain ontology for crowdsourcing platforms, specifically for microtasks.

3.1 Core Concepts

We define the requirements of the ontology in terms of the core concepts used by major crowdsourcing marketplaces. Existing literature in human computation and crowdsourcing has mainly described the concepts related to platforms design in the form of taxonomies [6, 8]. By comparison, our objective is to define the ontology in terms of what actions people can do for crowdsourcing systems and what human characteristics they need to perform those actions. Therefore, the following concepts constitute the main requirements of the ontology:

- **Task:** This concept is commonly used in the literature and crowdsourcing platforms to describe a unit of work to be performed by people in the crowd [2, 4]. Sometimes complex tasks are divided into smaller simple tasks to increase crowd participation [9].
- **Action:** The cognitive or psychomotor action or activity that leads towards the completion of a task [10]. A task can include one or more actions, for instance an audio transcription task includes activities of listening and writing.
- **User:** Commonly described as "worker" in crowdsourcing marketplaces due to the monetary payments earned by users [9]. However other crowdsourcing systems like Wikis and question answering systems used the concept of user to describe contributors.
- **Reward:** The concept of reward is popular in crowdsourcing marketplaces. However the existing literature considers this as a core concept related to the motivation of people in the crowd [9, 11, 12]. Although monetary rewards are common in marketplaces other motivating factors such as altruism, fun, learning, and reputation are also considered rewards.
- **Capability:** The human ability, knowledge, or skill that allows a user to perform the necessary actions for task completion [4, 13]. Availability and location of a person may include the requirements for some tasks.

3.2 Existing Ontologies

We have mapped the concepts described in ontology requirements with existing ontologies such as FOAF[8], SIOC[9], HRMO[10], PIMO[11], and TMO[12]. Table 2 shows how classes in the existing ontologies map with the concepts required for the ontology.

[8] http://www.foaf-project.org/

[9] http://sioc-project.org/ontology

[10] http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/ontologies/99-hrmontology

[11] http://www.semanticdesktop.org/ontologies/pimo/

[12] http://www.semanticdesktop.org/ontologies/2008/05/20/tmo/

Table 2. Mapping of concepts described in the ontology requirements with existing taxonomies and ontologies.

Concept	**PIMO**	**TMO**	**HRM-O**	**FOAF**	**SIOC**
Task	Task	Task			
Action					
User	Person		Job Seeker	Person	UserAccount
Reward			Compensation		
Reputation					
Money			Salary		
Fun					
Altruism					
Learning					
Capability					
Location			Location		
Skill		Skill			
Knowledge					
Ability			Ability		
Availability			Interval		

The *Personal Information Management Ontology* (PIMO) and *Task Management Ontology* (TMO) model tasks and skills of users of semantic desktops. The mappings with other ontologies or vocabularies can be defined using *owl:equivalentClass* and *owl:equivalentProperty* in web ontology language (OWL). Standard reasoning engines can be used to carry out the mappings for the instances of mapped ontologies. The coverage gap highlighted in Table 2 underlines the need for a separate ontology for human tasks, actions, rewards, and capabilities in crowdsourcing systems.

4 SLUA Ontology

The *Semantically Linked Users and Actions* (SLUA) ontology contains 5 main classes and 10 sub-classes that describe users and tasks in crowdsourcing systems. In the previous section we have identified the main concepts found in literature for describing the tasks and users in crowdsourcing systems. These concepts form the set of core classes in the SLUA ontology, as shown in Figure 1. Although similar concepts are captured by other ontologies, it is the relationships, class hierarchy, and properties of these concepts that are unique to SLUA. In the rest of this section the classes and their relationships are described in more detail.

4.1 Main Classes

The list of classes in the SLUA ontology are

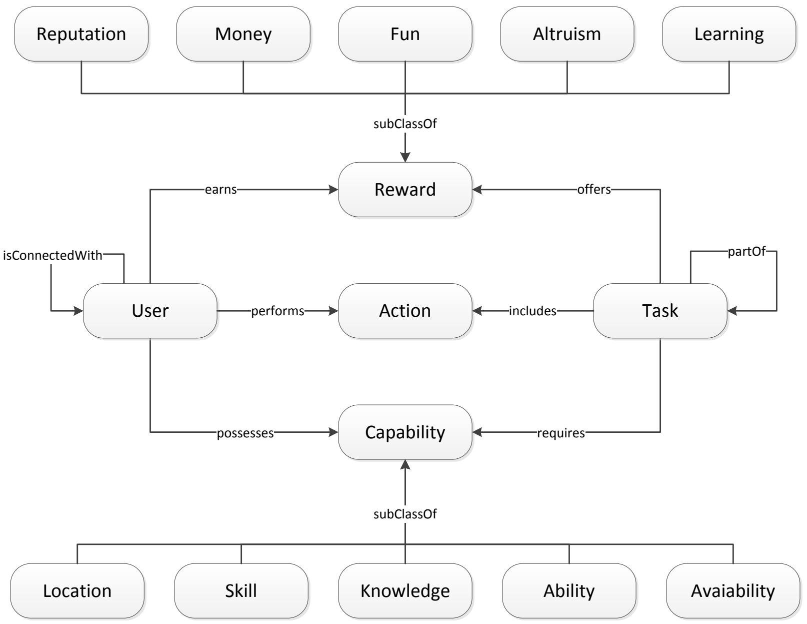


Fig. 1. Overview of the 5 main classes and 10 sub-classes defined in the SLUA ontology

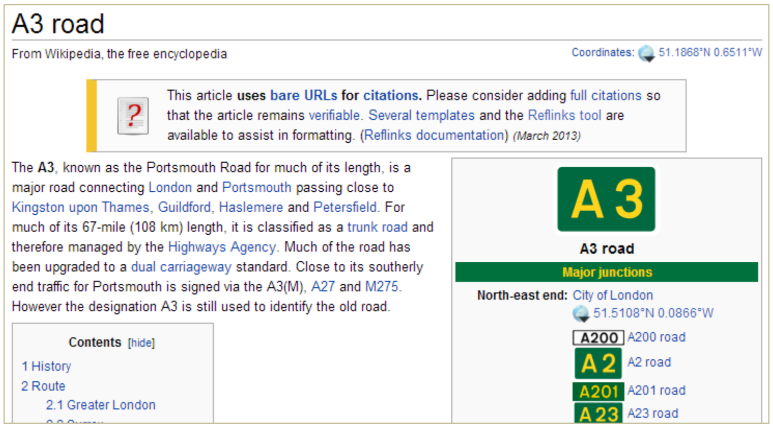
- **Action** class represents a specific act that is performed by the members of the crowd. An action can be cognitive or physical. For example, the comparison of two images involves a cognitive action from user.
- **Task** defines the unit of work resulting in a desired outcome that is assigned to the members of the crowd. A task may require one or more actions to produce the outcome. Therefore a task at the lowest level is composed of actions. The *Task* class has composition relationship with itself because complex tasks can be broken down into small simple tasks.
- **User** is the class that describes the human contributor in crowdsourcing. The user serves as an intelligent agent that is able to perform actions for successful completion of assigned tasks.
- **Reward** is associated with a task as the incentive for the human contribution. As noted earlier currently there are five types of reward classes:
  - **Fun** class represents rewards involving entertainment value such as games.
  - **Money** class represents monetary rewards.
  - **Fame** class represents rewards that benefit people in terms of recognition such as top contributors in Wikipedia.
  - **Altruism** class represents rewards involving social good.

  - **Learning** class represents rewards resulting in personal improvement in skill or knowledge.
- **Capability** defines the human characteristics that are necessary for performing a task. For instance one system might specify a user's location capability while another system utilizes this description to assign tasks relevant to the same location. There are five sub-classes defining different capabilities:
  - **Ability** class represents the stable capacity of users to engage in a specific behavior.
  - **Knowledge** class represents a body of information accumulated by users through education or experience.
  - **Skill** class represents the proficiency of a user in performing a task. Skill is acquired through training and practice.
  - **Location** class represents the specific place where a user is or will be physically present. This type of capability enables crowd contributions that are related to a physical place.
  - **Availability** class represents the time interval or time instant during which a user can perform a task.

4.2 Important Properties

This sub-section describes the properties of SLUA concepts that are important for extracting meaning from classes

- **domain:** A domain definition applies to most of the classes defined above. This property can be helpful for domain specific algorithms. A common categorization system could be used to specify domains in general crowdsourcing systems. However for specific areas purpose built taxonomies defined can be more effective.
- **offers:** This property defines the relationship of *Reward* with *Task*. For example some tasks might be rewarded with money. By comparison a user who is interested in a particular reward can be described with the **earns** property.
- **requires:** A *Task* can define requirements of one or more human capabilities using this property. By contrast a *User* can be described by having similar capabilities using the **possesses** property.
- **includes:** A *Task* can define one or more actions that a *User* **performs** for generating the desired outcome of a task.
- **isPartOf:** A complex *Task* can be decomposed into small manageable tasks. Therefore this property helps is describing the composition relationship between tasks.
- **hasDeadline:** This property can be used to specify time limitations of a *Task*, which is specifically important for real-time systems employing crowds.
- **isConnectedWith:** In the context of social networks, users are connected with other users through various relations. This property captures the network structure of users to enable social network based analysis of actions and users. For example the network structure can be exploited to recommend actions to neighbor nodes in a network.



A3 road

From Wikipedia, the free encyclopedia

Coordinates: 51.1868°N 0.6511°W

This article **uses bare URLs for citations.** Please consider adding full citations so that the article remains verifiable. Several templates and the Reflinks tool are available to assist in formatting. (Reflinks documentation) *(March 2013)*

The **A3**, known as the Portsmouth Road for much of its length, is a major road connecting London and Portsmouth passing close to Kingston upon Thames, Guildford, Haslemere and Petersfield. For much of its 67-mile (108 km) length, it is classified as a trunk road and therefore managed by the Highways Agency. Much of the road has been upgraded to a dual carriageway standard. Close to its southerly end traffic for Portsmouth is signed via the A3(M), A27 and M275. However the designation A3 is still used to identify the old road.

Contents [hide]

1 History

2 Route

2.1 Greater London

Fig. 2. Example of an article with a cleanup task that suggests users to add verifiable references to meet the quality standards of Wikipedia.

There are also domains specific properties that can be used to describe SLUA instance, as exemplified later in this paper.

5 Using SLUA

The core objective of SLUA is to provide a simple language to describe human tasks in crowdsourcing platforms to facilitate connectivity of tasks with users who can perform them. The SLUA ontology enables exchange of information on tasks, actions, users, rewards, and capabilities across crowdsourcing platforms. In the rest of this section we illustrate the use of SLUA in describing the semantics of tasks and users in different crowdsourcing systems.

5.1 Describing Tasks

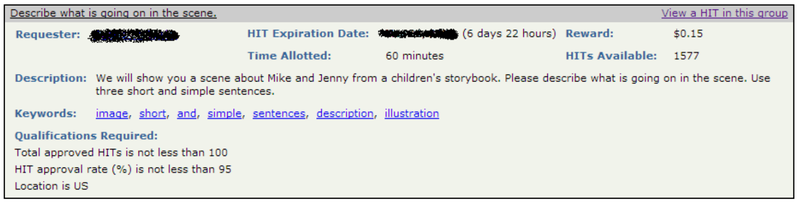
Collaborative Information Management. Wikipedia is a large collection of textual articles edited collaboratively by users on the Web. Articles in Wikipedia are routinely tagged for cleanup tasks due to issues with content or style. These tasks include adding new references, revising articles, merging sections, etc. For example, Figure 2 shows the alert message for an article about the "A3 road" in the City of London. The message suggests an action is required to remedy the quality issue with the article.

The alert message of a Wikipedia article and associated task can be described in *Resource Description Framework* (RDF) format using SLUA ontology. This allows machine readable access to human actions for improving quality of content in Wikipedia. The following code gives the example of the Wikipedia cleanup task converted to an instance of the *Task* class in SLUA.

```
<http://www.wikipedia.org/wiki/A3_road/tasks/1> a
slua:Task ;
    rdfs:label "Please consider adding full citations to
the Wikipedia article";
    slua:requires [
        a slua:Location;
        slua:locatedIn
<http://live.dbpedia.org/resource/London> ];
    slua:requires [
        a slua:Knowledge;
        slua:locatedIn
<http://live.dbpedia.org/resource/Roads> ];
    slua:offers [
        a slua:Reward;
        a slua:Reputation;
        slua:amount "1 star" ];
    slua:includes [
        a slua:Action
        rdfs:label "Wiki page edit"] .
```

Online Crowdsourcing Marketplace. Amazon Mechanical Turk is an online marketplace where requesters submit human intelligence task (HIT) to be performed by workers (i.e. users) in return for small amounts of money. Figure 3 shows example of task requiring users to describe a video with short sentences. Using SLUA the task can be described as an instance of *Task* class. The task requires two human capabilities; the capability of *Location* having *locatedIn* property with value "United States" and the capability of *Availability* with *availableFor* property with value "60" minutes. By performing the task workers can earn *Reward* of type *Money* with *amount* property of value "$0.15".

Cyber-Physical System. Next generation building management systems [14] involving human-in-the-loop for performing physical actions in the environment [15]. These systems ask occupants to perform environmental actions such as closing windows for reducing energy usage. A human action in building energy management serves as


Describe what is going on in the scene.
View a HIT in this group

Requester: [redacted]
HIT Expiration Date: [redacted] (6 days 22 hours)
Reward: $0.15
Time Allotted: 60 minutes
HITs Available: 1577

Description: We will show you a scene about Mike and Jenny from a children's storybook. Please describe what is going on in the scene. Use three short and simple sentences.

Keywords: image, short, and, simple, sentences, description, illustration

Qualifications Required:
Total approved HITs is not less than 100
HIT approval rate (%) is not less than 95
Location is US

Fig. 3. Example of human intelligence task (HIT) on Amazon Mechanical Turk

another use case for use of SLUA ontology. In this case the window closing action can be described as *Task*, which requires location capability; the capability of *Location* having *locateNear* property with value “Room A1” and having *locationTime* property with value “10:00PM”.

5.2 Describing Users

Similar to the description of tasks, the users of crowdsourcing platforms can be described using SLUA. Users can be described in terms of the actions they perform, the rewards they earn, and the capabilities they possess. The connection between various users can also be described to facilitate social network analysis. The following code gives an example of a Wikipedia user described with SLUA in RDF Turtle format.

```
<http://www.wikipedia.org/wiki/user/u0901> a slua:User .
    faof:name "Umair ul Hassan";
    slua:possess [
        a slua:Location;
        slua:locatedIn
<http://live.dbpedia.org/resource/London> ];
    slua:possess [
        a slua:Knowledge;
        slua:locatedIn
<http://live.dbpedia.org/resource/Roads> ];
    slua:earns [
        a slua:Reputation;
        slua:amount "4 star" ].
```

5.3 Leveraging Semantic Descriptions

Improving the routing of tasks to appropriate users is another objective of SLUA. In this regard the semantic descriptions of users and task can be used to perform the routing process. There are three major components of task routing system, as shown in Figure 4.

- **Task Modeling:** Uses SLUA to describe tasks. The capabilities of tasks can be discovered using methods such as cognitive task analysis [16].
- **Worker Profiling:** Uses SLUA to describe profiles of Users. The profiles can be generated using techniques such as expertise retrieval, behavior analysis, performance analysis, etc.
- **Task Routing:** Given task and user descriptions, this process involves finding suitability of user for a task. Depending on the tasks and users a variety of semantic similarity[13] approaches can be used of the purpose of matching.

[13] http://en.wikipedia.org/wiki/Semantic_similarity

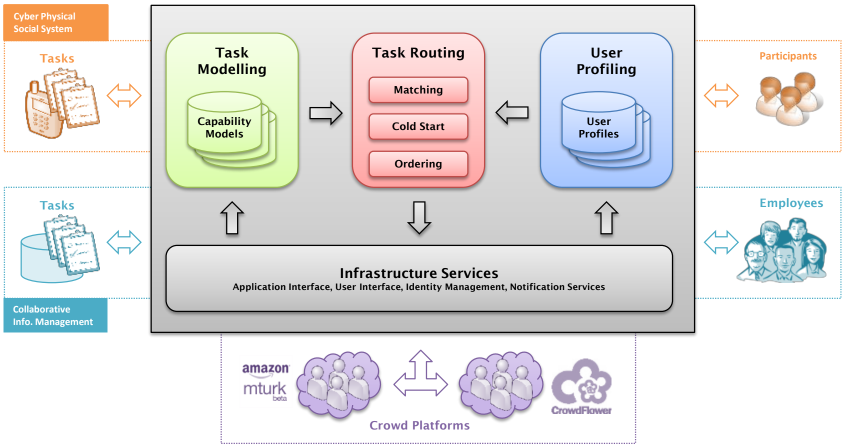


Fig. 4. Architecture of task routing system for heterogeneous tasks and users in crowdsourcing

6 Related Work

There has been considerable work on studying various dimensions of systems combining efforts of networked humans and computers. Malone et al. [1] described a framework of understanding working of a system based on collective intelligence. Doan et al. [2] discussed application of crowdsourcing to various domains. Quinn and Bederson [6] developed a taxonomy of human computation systems. Kearns [17] described tasks suited for social computing. These studies describe various aspects of human actions from a research perspective. By comparison this paper attempts to described actions and users for interoperability.

Bernstein et al. [13] called for the development of “social operating systems” for managing and allocating tasks to human resources at the global scale. Kittur et al. [9] highlight task assignment as the main research challenge for crowd work. Similarly, task routing has be defined as the fundamental aspect of human computation [4]. Ul Hassan etl al. have studied the relationship between user expertise and task routing in collaborative data quality management [18–20]. Diffallah et al. used social network profiles of users for assigning crowdsourcing tasks [21]. In this regard, SLUA provides a common language for the matching of tasks and users in crowdsourcing systems.

Existing ontologies, such as Personal Information Management Ontology [22] and Task Management Ontology [23], model some aspects of human actions and human capabilities. However these ontologies focus on task management from a desktop applications perspective. By comparison, SLUA specifies terms for crowdsourcing systems including rewards, capabilities, and actions.

7 Summary and Future Work

Semantically Linked Users and Actions is an initial step towards defining a lightweight ontology for describing tasks, actions, users, rewards, and capabilities in crowdsourcing platforms. This paper describes the core concepts and properties of SLUA ontology. This paper also gives example uses of SLUA to describe actions in different crowdsourcing scenarios. Future work includes the development of a prototype for exporting SLUA data from crowdsourcing platforms and developing a system that performs matchmaking between users and tasks using SLUA descriptions.

Acknowledgement. The work presented in this paper has been partially funded by Science Foundation Ireland Grant No. SFI/08/CE/I1380 (Lion- 2).

References

1. Malone, T.W., Laubacher, R., Dellarocas, C.N.: Harnessing Crowds: Mapping the Genome of Collective Intelligence, http://www.ssrn.com/abstract=1381502, (2009).
2. Doan, A., Ramakrishnan, R., Halevy, A.Y.: Crowdsourcing systems on the World-Wide Web. Communications of the ACM. 54, 86–96 (2011).
3. Schuler, D.: Social computing. Communications of the ACM. 37, 28–29 (1994).
4. Law, E., Ahn, L. von: Human Computation. Synthesis Lectures on Artificial Intelligence and Machine Learning. pp. 1–121 (2011).
5. Curry, E., Freitas, A., Riain, S.O.: The Role of Community-Driven Data Curation for Enterprises. In: Wood, D. (ed.) Linking Enterprise Data. pp. 25–47. Springer US, Boston, MA (2010).
6. Quinn, A.J., Bederson, B.B.: Human computation: a survey and taxonomy of a growing field. Proceedings of the 2011 annual conference on Human factors in computing systems - CHI ’11. pp. 1403–1412. ACM Press, New York, New York, USA (2011).
7. Uschold, M., Gruninger, M.: Ontologies and semantics for seamless connectivity. ACM SIGMOD Record. 33, 58 (2004).
8. Geiger, D., Seedorf, S., Nickerson, R.C., Schader, M., Nickerson, R.: Managing the Crowd : Towards a Taxonomy of Crowdsourcing Processes. (2011).
9. Kittur, A., Nickerson, J. V., Bernstein, M., Gerber, E., Shaw, A., Zimmerman, J., Lease, M., Horton, J.: The future of crowd work. Proceedings of the 2013 conference on Computer supported cooperative work - CSCW ’13. pp. 1301–1318. ACM Press, New York, New York, USA (2013).
10. Sakamoto, Y., Tanaka, Y., Yu, L., Nickerson, J. V: The Crowdsourcing Design Space. 346–355 (2011).
11. Simperl, E., Cuel, R., Stein, M.: Incentive-Centric Semantic Web Application Engineering. Synthesis Lectures on the Semantic Web: Theory and Technology. 3, 1–117 (2013).
12. Tokarchuk, O., Cuel, R., Zamarian, M.: Analyzing Crowd Labor and Designing Incentives for Humans in the Loop. IEEE Internet Computing. 45–51 (2012).
13. Bernstein, A., Klein, M., Malone, T.W.: Programming the global brain. Communications of the ACM. 55, 41 (2012).
14. Curry, E., Hasan, S., O’Riain, S.: Enterprise Energy Management using a Linked Dataspace for Energy Intelligence. Second IFIP Conference on Sustainable Internet and ICT for Sustainability. , Pisa, Italy (2012).

15. Crowley, D.N., Curry, E., Breslin, J.G.: Closing the Loop-From Citizen Sensing to Citizen Actuation. 7th IEEE International Conference on Digital Ecosystem Technologies. (2013).
16. Schraagen, J.M., Chipman, S.F., Shalin, V.L.: Cognitive Task Analysis. Taylor & Francis (2000).
17. Kearns, M.: Experiments in social computation. Communications of the ACM. 55, 56 (2012).
18. Ul Hassan, U., O'Riain, S., Curry, E.: Towards Expertise Modelling for Routing Data Cleaning Tasks within a Community of Knowledge Workers. Proceedings of the 17th International Conference on Information Quality. , Paris, France (2012).
19. Ul Hassan, U., O'Riain, S., Curry, E.: Effects of Expertise Assessment on the Quality of Task Routing in Human Computation. Proceedings of the 2nd International Workshop on Social Media for Crowdsourcing and Human Computation. , Paris, France (2013).
20. Ul Hassan, U., O'Riain, S., Curry, E.: Leveraging Matching Dependencies for Guided User Feedback in Linked Data Applications. Proceedings of the Ninth International Workshop on Information Integration on the Web. pp. 1–6. ACM Press, New York, New York, USA (2012).
21. Difallah, D.E., Demartini, G., Cudré-Mauroux, P.: Pick-a-crowd: tell me what you like, and i'll tell you what to do. Proceedings of the 22nd international conference on World Wide Web. pp. 367–374. International World Wide Web Conferences Steering Committee (2013).
22. Sauermann, L., Elst, L. van, Dengel, A.: PIMO - a Framework for Representing Personal Information Models. Proceedings of I-Semantics' 07. pp. 270–277. JUCS, Graz, Austria (2007).
23. Ong, E., Riss, U. V., Grebner, O., Du, Y.: Semantic Task Management Framework. Proceedings of I-KNOW '08. pp. 387–394. JUCS, Graz, Austria (2008).

Measuring Crowd Truth: Disagreement Metrics Combined with Worker Behavior Filters

Guillermo Soberón[1], Lora Aroyo[1], Chris Welty[2], Oana Inel[1], Hui Lin[3], and Manfred Overmeen[3]

[1] VU University, Amsterdam, The Netherlands,
guillelmo@gmail.com, l.m.aroyo@cs.vu.nl, oana.inel@vu.nl,
[2] IBM Research, New York, USA
cawelty@gmail.com,
[3] IBM Netherlands, Amsterdam, The Netherlands
hui.lin2013@nl.ibm.com, manfred.overmeen@nl.ibm.com

Abstract. When crowdsourcing gold standards for NLP tasks, the workers may not reach a consensus on a single correct solution for each task. The goal of Crowd Truth is to embrace such disagreement between individual annotators and harness it as useful information to signal vague or ambiguous examples. Even though the technique relies on disagreement, we also assume that the differing opinions will cluster around the more plausible alternatives. Therefore it is possible to identify workers who systematically disagree - both with the majority opinion and with the rest of their co-workers- as low quality or spam workers. We present in this paper a more detailed formalization of metrics for Crowd Truth in the context of medical relation extraction, and a set of additional filtering techniques that require the workers to briefly justify their answers. These explanation-based techniques are shown to be particularly useful in conjunction with disagreement-based metrics, and achieve 95% accuracy for identifying low quality and spam submissions in crowdsourcing settings where spam is quite high.

Keywords: crowdsourcing, disagreement, quality control, relation extraction

1 Introduction

The creation of gold standards by expert annotators can be a very slow and expensive process. When it comes to NLP tasks, like relation extraction, annotators have to deal with the ambiguity of the expressions in the text in different levels, frequently leading to disagreement between annotators. To overcome this, detailed guidelines for annotators are developed, in order to handle the different cases that have been observed, through practice, to generate disagreement. However, the process of avoiding disagreement has lead in many cases to brittleness and over generality in the ground truth, making it difficult to transfer annotated data between domains or to use the results for anything practical.

In comparison with expert generated ground truth, crowdsourcing gold standard can be a cheaper and more scalabe solution. Crowdsourced gold standards typically show lower overall κ scores [3], especially for complex NLP tasks such as relation extraction, since the workers perform small, simple (micro) tasks and cannot be relied on to read a long guideline document. Rather than eliciting an artificial agreement between workers, in [1] we presented "Crowd Truth", a crowdsourced gold standard technique that, instead of considering the lack of agreement something to be avoided, it is used as something informative from which characteristics and features of the annotated content may be inferred. For instance, a high disagreement for a particular sentence may be a sign of ambiguity in the sentence.

As the final Crowd Truth is a by-product of the different contributions of the members of the crowd, being able to identify and filter possible low quality contributors is crucial to reduce their impact on the overall quality of the aggregate result. Most of the existing approaches for detecting low quality contributions in crowdsourcing tasks are based on the assumption that for each task there is a single correct answer, enabling distance and clustering metrics to detect outliers [14] or using gold units [15], establishing an equivalency between disagreement with the majority and low quality contributions.

For Crowd Truth the initial premise is that there is not only one right answer, and the diversity of opinions is to be preserved. However, disagreement with the majority can still be used as a way to distinguish low quality annotators.For each task, it may be assumed that the workers answers will be distributed among the possible options, with the most plausible answers concentrating the highest number of workers, and the improbable answers being stated by none or very few workers. That way, workers whose opinions are different from those of the majority, are likely to find other workers with similar views over the issue. On the other hand, the answers of workers who complete the task randomly or without understanding the task or its content, tend to be not aligned with those of the rest. Hence, it would be possible to filter by identifying those workers who, not only disagree with the majority opinion of the crowd on a task basis, but whose opinions are systematically not shared by many of their peers. The initial definition of the content-based disagreement metrics was introduced by [1] to identify and filter low quality workers for relation extraction tasks, establishing metrics for the inter-worker agreement and the agreement with the crowd opinion.

While filtering workers by disagreement has showed to be an effective way of detecting low quality contributors, achieving high precision, we demonstrate that it is not sufficient to filter all the existing ones. We have extended the relation extraction task by asking the workers to provide a written justification for their answers, and the manual inspection of the results contained several instances of badly formed, incomplete or even random-text explanations, which can be securely attributed to low quality workers or even automated spam bots.

In order to complement the disagreement filters, we propose several ways to use the explanations provided by the contributors, to implement new low quality worker filters that extend and complement the recall of the disagreement filter.

2 Related Work

2.1 Disagreement

In the absence of gold standard, a different evaluation schemes can be used for worker quality evaluation. For instance, the results among workers can be compared and the agreement in their responses can be used as quality estimator.

As is well known [4], the frequency of disagreement can be used to estimate worker error probabilities. In [9] the computation of quality estimators for workers quality based on disagreement is proposed as part of a set of techniques to evaluate workers, along with confidence intervals for each one of this schemas; which allows to estimate the "efficiency" of each one of them.

A simpler method is proposed in [16], which assumes "plurality of answers" for a task, and estimates the quality of a worker based on the number of tasks for which a worker agrees with "the plurality answer" (i.e the one from the majority of the workers).

While these disagreement-based schemas do not rely on the assumption that there is only one single answer per task (thus, allowing room for disagreement between workers responses), they still assume a correlation between disagreement and low quality of the worker. Crowd Truth not only allows but **fosters** disagreement between the workers, as it is considered informative.

2.2 Filtering by explanations

As stated in [6], cheaters tend to avoid tasks that involve creativity and abstract thinking, and even for simple straightforward tasks, the addition of the non-repetitive elements discourage low quality contribution and automation of the task. Apart from the dissuasive element for spammers of introducing these non-repetitive elements in the task design, our work additionally tries to use this as a base for filtering once the task is completed.

Previous experiences [12] have shown that workers tend to provide good answers to open-ended questions when those are concrete, and response length can be used as an indicator of the participant engagement in the task.

2.3 Crowd Watson

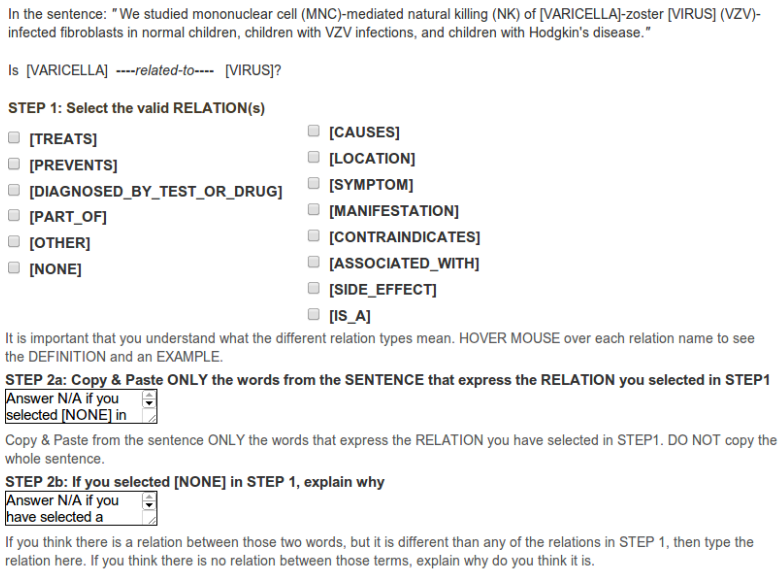
Watson [7] is an artificial intelligent system capable of answering questions posed in natural language designed by IBM. To build its knowledge base Watson was trained on a series of databases, taxonomies, and ontologies of publicly available data [10]. Currently, IBM Research aims at adapting the Watson technology for question-answering in the medical domain. For this, large amounts of training and evaluation data (ground truth medical text annotation) are needed, and the traditional ground-truth annotation approach is slow and expensive, and constrained by too restrictive annotation guidelines that are necessary to achieve good inter-annotator agreement, which result in the aforementioned over generalization.

The Crowd Watson project [11] implements the Crowd truth approach to generate a crowdsourced gold standard for training and evaluation of IBM Watson NLP components in the medical domain. Complementary to the Crowd truth implementation, and within the general Crowd Watson architecture, a gaming approach for crowdsourcing has been proposed [5], as a way to enhance engagement of the experts annotators.

Also, within the context of the Crowd Watson project, [8] has shown how the worker metrics initially set up for the medical domain can be adapted to other domains and tasks, such as event extraction.

3 Representation

CrowdFlower workers were presented sentences with the argument words highlighted and 12 relations (manually selected from UMLS [2]) as shown below in Fig 1; they were asked to choose all the relations from the set of 12 that related the two arguments in the sentence. They were also given the options to indicate that the argument words were not related in the sentence (NONE), or that the argument words were related but not by one of the 12 relations (OTHER). They were also asked to justify their choices by *selecting the words* in the sentence that they believed "signaled" the chosen relations or, in case they chose NONE or OTHER, provide the *rationale* for that decission.



In the sentence: "We studied mononuclear cell (MNC)-mediated natural killing (NK) of [VARICELLA]-zoster [VIRUS] (VZV)-infected fibroblasts in normal children, children with VZV infections, and children with Hodgkin's disease."

Is [VARICELLA] ----*related-to*---- [VIRUS]?

STEP 1: Select the valid RELATION(s)

□ **[TREATS]**
□ **[PREVENTS]**
□ **[DIAGNOSED_BY_TEST_OR_DRUG]**
□ **[PART_OF]**
□ **[OTHER]**
□ **[NONE]**

□ **[CAUSES]**
□ **[LOCATION]**
□ **[SYMPTOM]**
□ **[MANIFESTATION]**
□ **[CONTRAINDICATES]**
□ **[ASSOCIATED_WITH]**
□ **[SIDE_EFFECT]**
□ **[IS_A]**

It is important that you understand what the different relation types mean. HOVER MOUSE over each relation name to see the DEFINITION and an EXAMPLE.

STEP 2a: Copy & Paste ONLY the words from the SENTENCE that express the RELATION you selected in STEP1

Answer N/A if you selected [NONE] in

Copy & Paste from the sentence ONLY the words that express the RELATION you have selected in STEP1. DO NOT copy the whole sentence.

STEP 2b: If you selected [NONE] in STEP 1, explain why

Answer N/A if you have selected a

If you think there is a relation between those two words, but it is different than any of the relations in STEP 1, then type the relation here. If you think there is no relation between those terms, explain why do you think it is.

Fig. 1. Relation Extraction Task Example

Note, that the process and the choices for setting up the annotation template is out of scope for this paper. Relation extraction task is part of the larger crowdsourcing framework, Crowd-Watson, which defines the input text, the templates and the overall workflow [1]. In this paper we only focus on the representation and analysis of the collected crowdsourced annotations.

The information gathered from the workers is represented using vectors in which components are all the relations given to the workers (including the choices for NONE and OTHER). All metrics are computed from three vector types:

1. *worker-sentence vector* $V_{s,i}$ The result of a single worker annotating a single sentence. For each relation that the worker annotated in the sentence, there is a 1 in the corresponding component, otherwise a 0.
2. *sentence vector* V_s The vector sum of the worker-sentence vectors for each sentence $V_s = \sum_i V_{s,i}$
3. *relation vector* R_i A unit vector in which only the component for relation i is 1, the rest 0.

We collect two different kinds of information: the annotations and the explanations about the annotations (i.e the *selected words* that signal the chosen relation, or their *rationale* for selecting NONE or OTHER).

We try to identify behaviour that can be associated with low quality workers from the perspective of these two domains: *disagreement metrics* rely on the content of the annotations to identify workers that systematically disagree with the rest; *explanation filters* aim at identifying individual behaviours that can be attributed to spammers or careless workers.

4 Disagreement metrics

As with the semiotic triangle [13], there are three parts to understanding a linguistic expression: the sign, the thought or interpreter, the referent. We instrument the crowdsourcing process in three analogous places: the micro-task, for the relation extraction case this is a sentence; the workers, who interpret each sentence; the task semantics, in the case of relation extraction this is the intended meaning of the relations.

4.1 Sentence Metrics

Sentence metrics are intended to measure the quality of sentences for the relation extraction task. These measures are our primary concern, we want to provide the highest quality of training data to machine learning systems.

Sentence-relation score is the core crowd truth metric for relation extraction, it can be viewed as the probability that the sentence expresses the relation. It is measured for each relation on each sentence as the cosine of the unit vector for the relation with the sentence vector: $srs(s,r) = cos(V_s, R_r)$

The relation score is used for training and evaluation of the relation extraction system. This is a fundamental shift from the traditional approach, in which sentences are simply labelled as expressing, or not, the relation, and presents new challenges for the evaluation metric and especially for training.

Sentence clarity is defined for each sentence as the max sentence-relation score for that sentence: $scs(s) = max_r(srs(s, r))$

If all the workers selected the same relation for a sentence, the max relation score will be 1, indicating a clear sentence.

Sentence clarity is used to weight sentences in training and evaluation of the relation extraction system, since annotators have a hard time classifying them, the machine should not be penalized as much for getting it wrong in evaluation, nor should it treat such training examples as exemplars.

4.2 Worker Metrics

Worker metrics are primarily to establish worker quality; low quality workers and spammers should be eliminated as they contribute only noise to the disagreement scores, and high quality workers may get paid more as an incentive to return. We investigated several dimensions of worker quality for the relation extraction task:

Number of annotations per sentence is a worker metric indicating the average number of different relations per sentence used by a worker for annotating a set of sentences. Unambiguous sentences should ideally be annotated with one relation, and generally speaking each worker interprets a sentence their own way, but a worker who consistently annotates individual sentences with multiple relations usually does not understand the task.

Worker-worker agreement is the asymmetric pairwise agreement between two workers across all sentences they annotate in common:

$$wwa(w_i, w_j) = \frac{\sum_{s \in S_{i,j}} RelationsInCommon(w_i, w_j, s)}{\sum_{s \in S_{i,j}} NumAnnotations(w_i, s)}$$

where $S_{i,j}$ is the subset of all sentences S annotated by both workers w_i and w_j, $RelationsInCommon(w_i, w_j, s)$ is the number of identical annotations (relations selected) on a sentence between the two workers, and $NumAnnotations(w_i, s)$ is the number of annotations by a worker on a sentence.

Average worker-worker agreement is a worker metric based on the average worker-worker agreement between a worker and the rest of workers, weighted by the number of sentences in common. While we intend to allow disagreement, it should vary by sentence. Workers who consistently disagree with other workers usually do not understand the task:

$$avg\_wwa(w_i) = \frac{\sum_{j \neq i} |S_{i,j}|.wwa(w_i, w_j)}{\sum_{i \neq j} |S_{i,j}|}$$

Worker-sentence similarity is the vector cosine similarity between the annotations of a worker and the aggregated annotations of the other workers in a sentence, reflecting how close the relation(s) chosen by the worker are to the opinion of the majority for that sentence. This is simply $wss(w_i, s) = cos(V_s - V_{s,i}, V_{s,i})$

Worker-sentence disagreement is a measure of the quality of the annotations of a worker for a sentence. It is defined, for each sentence and worker, as the difference between the Sentence Clarity (q.v. above) for the sentence and the worker sentence similarity for that sentence: $wsd(w_i, s) = scs(s) - wss(w_i, s)$. Workers who differ drastically from the most popular choices will have large disagreement scores, workers who agree with the most popular choice will score 0.

The intuition for using the difference from the clarity score over the cosine similarity, as originally proposed in [1], is to capture worker quality on a sentence compared to the quality of the sentence itself. In uni-modal cases, e.g. where a sentence has one clear majority interpretation, the cosine similarity works well, but in the case where a sentence has a bimodal distribution, e.g. multiple popular interpretations, the worker's cosine similarity will not be very high even for those that agree with one of the two most popular interpretations, which seems less desireable.

Average worker-sentence disagreement is a worker metric based on the average worker-sentence disagreement score across all sentences, $avg\_wsd(w_i) = \frac{\sum_{s \in S_i} wsd(w_i, s)}{|S_i|}$ where S_i is the subset of all sentences annotated by worker w_i.

The worker-worker and worker-sentence scores are clearly similar, they both measure deviation from the crowd, but they differ in emphasis. The *wsd* metric simply measures the average divergence of a worker from the crowd on a sentence basis, someone who tends to disagree with the majority will have a low score. For *wwa*, workers who may not always agree with the crowd on a sentence basis might be found to agree with a group of people that disagree with the crowd in a similar way, and would have a low score. This could reflect different cultural or educational perspectives, as opposed to simply a low quality worker.

4.3 Relation Metrics

Relation clarity is defined for each relation as the max sentence-relation score for the relation over all sentences:

$rcs(r) = max_s(srs(s, r))$

If a relation has a high clarity score, it means that it is at least possible to express the relation clearly. We find in our experiments that a lot of relations that exist in structured sources are very difficult to express clearly in language, and are not frequently present in textual sources. Unclear relations may indicate unattainable learning tasks.

5 Explanation filters

In [1] we showed results using the worker metrics to detect low quality workers. In order to evaluate our results, we had workers justify their answers. The explanations of the annotation tasks are not strictly necessary for the crowd truth data, and represent additional time and therefore cost to gather. In this section we analyze the value of this information.

We examined whether this additional effort dissuaded workers from completing the task. Two different implications are to be distinguished for this circumstance: one positive, by driving away low quality workers or spammers -whose main objective is to maximize its economic reward with the minimum possible effort-; and one negative, as it may induce some good contributors to choose easier, less demanding tasks. In order to prevent this, it might be necessary to increase the economic reward to make up for the extra effort, so, at the end, the addition of explanations implies an increase in the task price. And, finally, we want to test whether the explanations -apart from preventing low quality workers to complete the task- may contain information that it is useful for detecting low quality workers.

Apart from the presence of explanations, another variable to take into account for spam detection is the *channel* of the workers. CrowdFlower has over 50 labor channel partners or external labor of workers, such as Amazon Mechanical Turk and TrialPay, which can be used (individually or combined) to run crowdsourcing processes. Our intuition was that different channels have different spam control mechanisms, which may redound in different spammer ratios, depending on the channel.

To explore these variables, we set up an experiment to annotate the same 35 sentences, over different configurations:

1. Without explanations, using workers from multiple Crowdflower channels
2. Without explanations, using workers from Amazon Mechanical Turk (AMT)
3. With explanations, using workers from multiple Crowdflower channels
4. With explanations, using workers from AMT

Note that AMT was among the multiple channels used on 1 and 3, but the presence of workers from AMT was minority.

By comparing the pairs formed by 1 and 2, and 3 and 4, we can test whether the channel has any influence in the low quality worker ratio. Likewise, the pairs formed by 1 and 3, and 2 and 4, can be used to test the influence of the explanations, independently of the channel used.

We collected 18 judgments per sentence (for a total of 2522 judgements), and workers were allowed to annotate a maximum of 10 different sentences. The number of unique workers per batch was comprehended between 67 and 77 workers.

In the results we observed that the time to run the task using multiple channels was significantly lower than doing so only on AMT, independently of whether the explanations where required or not. The time invested on annotating a sentence of the batch was substantially lower, on average, when explanations were not required.

The number of workers labelled as possible low quality workers by the disagreement filters was low, and more or less was kept within the same range for the four batches (between 6 and 9 filtered workers per batch); so we cannot infer whether including explanations discourages low quality workers from working in it.

However, manual exploration of the annotations revealed four patterns that may be indicative of possible spam behaviour:

1. **No valid words** (No Valid in Table 1) were used, either on the explanation or in the selected words, using instead random text or characters.
2. Using the **same text for both the explanation and the selected words** (Rep Resp in Table 1). According to the task definition, both fields are exclusive: either the explanation or the selected words that indicate the rationale of the decision are to be provided, so filling in both may be due bad understanding of the task definitions. Also, both are semantically different reasons, so it is unlikely that the same text is applicable for both.
3. Workers that **repeated the same text** (Rep Text in Table 1) for all their annotations, either justifying their choice using the exact same words or selecting the same words from the sentence.
4. **[NONE] and [OTHER] used with other relations** (None/Other in Table 1). None and Other are intended to be exclusive: according to the task definition, by selecting them the annotator is stating that none of the other relations is applicable for the sentence. Hence, it is semantically incorrect to choose [NONE] or [OTHER] in combination with other(s) relations, and doing so may reflect a bad understandment of the task definition.

The degree to which these patterns may indicate spam behaviour is different: in most cases, "No valid words" is a strong indicator of a low quality worker, while a bad use of [NONE] or [OTHER] may be the reflection of a bad understanding of the task (i.e. when should one text box be filled and when the other), rather than a bad worker.

Chan.	**Disag. Filters** (# Spam)	**Explanation filters** # Spam - (% Overlap w/ disagr. filters)				**# Spam exclusively detected by exp. filters**
		None / Other	**Rep Resp**	**Rep Text**	**No Valid**	
Multiple	9	7 (29%)	14 (29%)	3 (33%)	11 (36%)	18
AMT	6	9 (22%)	2 (0%)	2 (50%)	1 (0%)	11

Table 1. Results from 35 Sentences with explanation-based filters

Table 1 contains an overview of the number of occurrences in the batches with explanations of each of the previous patterns. For each pattern, the percentage of workers identified as low quality workers is indicated. This percentage and the last column -which indicates the number of workers for which at least one of the low quality patterns have been observed but are not labelled as low quality by the disagreement filters- shows that there is little overlap between these patterns and what the disagreement filters considers low quality "behaviour". Therefore, it seems reasonable to further explore the use of this patterns as "explanation filters" for low quality workers. Also, the number of potential low quality workers according to the spam patterns, seems bigger when the task is run on multiple channels rather than only on AMT. This observation cannot be considered conclusive, but it seems reasonable to explore it further.

6 Experiments

We designed a series of experiments to gather evidence in support of our hypothesis that the disagreement filters may not be sufficient and that the explanations can be used to implement additional filters to improve the spam detection.

6.1 Data

The data for the main experiments consist of two different sets of 90 sentences. The first set (Experiment 2 or EXP2) is annotated only by workers from Amazon Mechanical Turk (AMT), and the second (Experiment 3 or EXP3) is annotated by workers from multiple channels among those offered by Crowdflower (including AMT, though the AMT workers were a minority).

To optimize the time and worker dynamics we split the 90 sentences sets in batches of 30 sentences. The batches of the first set were run on three different days, and the batches of the second were all run on the same day. Workers were not allowed to annotate more than 10 sentence in the first set, and no more than 15 in the second. We collected 450 judgments (15 per sentence) in each batch (for a total of 1350 per set), from 143 unique workers in the first set and 144 in the second.

From our previous experiences, judgements from workers who annotated two or fewer sentences were uninformative, so we have removed these leaving 110 and 93 workers and a total of 1292 and 1302 judgements on each set.

We have manually gone through the data and identified low quality workers from their answers. 12 workers (out of 110) were identified as low quality workers for EXP2 and 20 (out of 93) for EXP3. While all the spammers on EXP2 were identified as such by the disagreement filters, only half of the low quality workers in EXP3 were detected.

Also it is important to notice that the number of annotations by workers identified spammers is much higher for EXP3 (386 out of 1291, 30%) than for EXP2 (139 out of 1302, 11%).

6.2 Filtering low quality workers

In this section, we address our hypotheses by, first, describing disagreement performance for EXP3, it is shown how it is not sufficient by itself; and, second, showing how the explanation filters are informative and disjoint from the disagreement filters (they indicate something, and that "something" is different from what disagreement points to).

A sense of the different disagreement metrics in detecting low quality workers is shown in Figure 2 and 3. Each metric is plotted against overall accuracy at different confidence thresholds, on each experiment. Clearly, the overall accuracy of the disagreement metrics is lower for EXP3. While it is possible to achieve a 100% accuracy for EXP2 by linearly combinining the disagreement metrics, only 89% is achieved for EXP3 by this means.

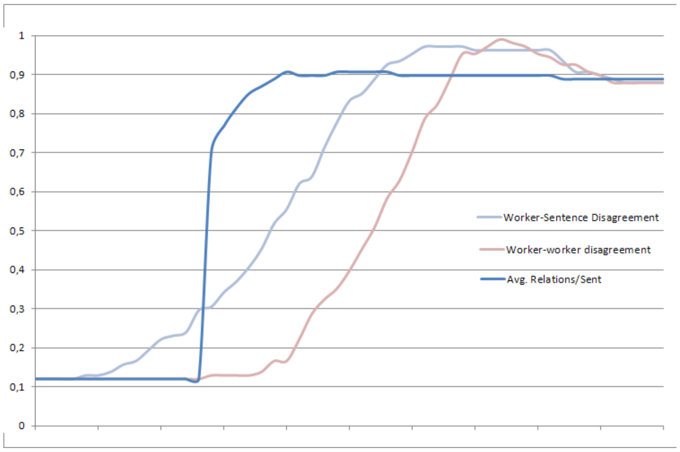


Fig. 2. Accuracy of worker metrics for predicting low quality workers at all thresholds, for Experiment 2

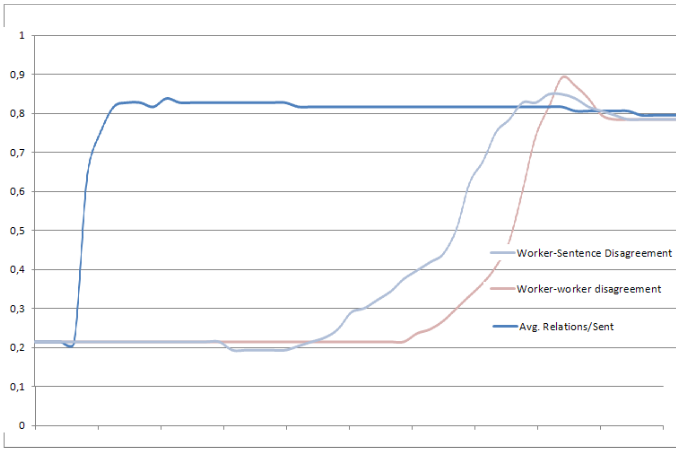


Fig. 3. Accuracy of worker metrics for predicting low quality workers at all thresholds, for Experiment 3

In order to make up for this, we analyzed the explanations filters, exploring whether they provide some information about possible spammer behaviour that it is not already contained in the disagreement metrics. The explanation filters are not very effective by themselves: their recall value is pretty low (in all cases, below 0.6), and it is not substantially improved by combining them.

The tables 2 and 3 present an overview of the workers identified as possible spammers by each filter, reflecting the intersections and differences between the disagreement filters and the explanation filters.

Note that we analyze the experiments both on a "job" basis, and on an aggregate "Experiment" basis. This displays how jobs are more or less "homogeneous" (for instance, that one of them is not clearly biased in one particular batch, therefore, biasing the aggregated experiment). However, for filtering purposes, we treat the experiments as atomic units.

Exp.	**Disag. Filters** (# Spam)	**Explanation filters** # Spam - (% Overlap w/ disagr. filters)				**# Spam exclusively detected by exp. filters**
		None / Other	**Rep Resp**	**Rep Text**	**No Valid**	
Batch 2.1	8	5 (40%)	2 (100%)	4 (75%)	0	4
Batch 2.2	6	3 (33%)	3 (66%)	0	0	3
Batch 2.3	5	2 (0%)	4 (25%)	0	1 (0%)	6
Total Exp 2	**12**	**10 (20%)**	**7 (43%)**	**4 (75%)**	**1 (0%)**	**14**

Table 2. Filters Overview - Experiment 2 (AMT)

It can be observed how the overlap (i.e. the number of workers identified as possible spammers by two different filters) between the disagreement filters and each of the explanation filters is not really significative.

Exp.	**Disag. Filters** (# Spam)	**Explanation filters** # Spam - (% Overlap w/ disagr. filters)				**# Spam exclusively detected by exp. filters**
		None / Other	**Rep Resp**	**Rep Text**	**No Valid**	
Batch 3.1	4	6 (33%)	7 (57%)	5 (20%)	6 (17%)	13
Batch 3.2	4	11 (18%)	0	7 (14%)	6 (33%)	15
Batch 3.3.	6	8 (37.5%)	0	5 (60%)	1 (0%)	8
Total Exp 3	**10**	**22 (18%)**	**8 (37%)**	**14 (36%)**	**12 (42%)**	**30**

Table 3. Filters Overview - Experiment 3 (Multiple channels)

On the other hand, the number of workers identified as possible spammers exclusively by the explanation filters is quite big for the EXP3. Not only it's higher than for EXP2, but also in comparison with the number of workers filtered by the disagreement filters. This is coherent with the manual identification of spammers, which revealed 26 spammers.

7 Results and future work

By linearly combining the filters, we have obtained a classifier with 95% accuracy and F-measure 0.88, improving the disagreement-only filtering (88% accuracy and F-measure 0.66) for EXP3. More data is needed to improve and rigorously validate this approach, but this initial results are already promising.

This linear combination of filters serves to the purpose of complementing disagreement filters with explanation filters. In future work, we will further explore different ways of combining these filters to improve quality, such as bagging.

For the current implementation, we have omitted the differences in the prediction power of each of the explanation filters, when it can be resonably assumed that they are not equally good indications of spam behaviour. It is also worth considering using a boosting approach to improve this.

Also, disagreement filters may be complemented by other kinds of information. For instance, for EXP3, the workers completing the task come from different channels. In future work, we will further explore whether the worker provenance is a significative toward low quality detection.

While sentence and worker metrics have proven to be informative, the available data is not sufficient to reach similar conclusions for the relation metrics, as the different relations are unevenly represented. We will try to collect more data in order to further explore this metrics.

8 Conclusions

We presented formalizations of sentence and worker metrics for Crowd Truth, and showed how the worker metrics could be used to detect low quality workers. We then introduced a set of explanation-based filters based on workers justification of their answers, and we ran experiments on various crowdsourcing "channels".

The conducted experiments seem to indicate that, when in presence of a small number of low quality **annotations**, disagreement filters are sufficient to preseve data quality. On the other hand, in the presence of a higher number of low quality annotations, the effectivity of disagreement filters diminishes, and are not enough to detect all the possible low quality contributions.

We have showed how the explanations provided by the workers about their answers can be used to identify patterns that can reasonably associated with spamming or low quality annotation behaviours. We used these these patterns combined with the worker metrics to detect low quality workers with 95% accuracy in a small cross-validation experiment.

References

1. Lora Aroyo and Chris Welty. Crowd truth: Harnessing disagreement in crowdsourcing a relation extraction gold standard. In *Proc. Websci 2013*. ACM Press, 2013.

2. Olivier Bodenreider. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research*, 32(suppl 1):D267–D270, 2004.
3. Jacob Cohen et al. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.
4. Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied Statistics*, pages 20–28, 1979.
5. Anca Dumitrache, Lora Aroyo, Chris Welty, and Robert-Jan Sips. Dr. Detective: combining gamificationtechniques and crowdsourcing to create a goldstandard for the medical domain. Technical report, VU University Amsterdam, 2013.
6. Carsten Eickhoff and Arjen P. de Vries. Increasing cheat robustness of crowdsourcing tasks. *Inf. Retr.*, 16(2):121–137, April 2013.
7. David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty. Building watson: An overview of the deepqa project. *AI Magazine*, 31:59–79, 2010.
8. Oana Inel, Lora Aroyo, Chris Welty, and Robert-Jan Sips. Exploiting Crowdsourcing Disagreement with Various Domain-Independent Quality Measures. Technical report, VU University Amsterdam, July 2013.
9. Manas Joglekar, Hector Garcia-Molina, and Aditya Parameswaran. Evaluating the crowd with confidence. 2012.
10. Aditya Kalyanpur, BK Boguraev, Siddharth Patwardhan, J William Murdock, Adam Lally, Chris Welty, John M Prager, Bonaventura Coppola, Achille Fokoue-Nkoutche, Lei Zhang, et al. Structured data and inference in DeepQA. *IBM Journal of Research and Development*, 56(3.4):10–1, 2012.
11. Hui Lin. Crowd Watson: Crowdsourced Text Annotations. Technical report, VU University Amsterdam, July 2013.
12. C. Marshall and F. Shipman. Experiences surveying the crowd: Reflections on methods, participation, and reliability. In *Proc. Websci 2013*. ACM Press, 2013.
13. C.K. Ogden and I. A. Richards. The meaning of meaning: A study of the influence of language upon thought and of the science of symbolism. *8th ed. 1923. Reprint New York: Harcourt Brace Jovanovich*, 1923.
14. Vikas C. Raykar and Shipeng Yu. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *J. Mach. Learn. Res.*, 13:491–518, March 2012.
15. Cristina Sarasua, Elena Simperl, and Natalya Fridman Noy. Crowdmap: Crowdsourcing ontology alignment with microtasks. In *International Semantic Web Conference (1)*, pages 525–541, 2012.
16. Petros Venetis and Hector Garcia-Molina. Quality control for comparison microtasks. In *Proceedings of the First International Workshop on Crowdsourcing and Data Mining*, pages 15–21. ACM, 2012.

Crowdsourced Entity Markup

Lili Jiang, Yafang Wang, Johannes Hoffart, Gerhard Weikum

Max Planck Institute for Informatics
Saarbruecken, Germany
{ljiang,ywang,jhoffart,weikum}@mpi-inf.mpg.de

Abstract. Entities, such as people, places, products, etc., exist in knowledge bases and linked data, on one hand, and in web pages, news articles, and social media, on the other hand. Entity markup, like Named Entities Recognition and Disambiguation (NERD), is the essential means for adding semantic value to unstructured web contents and this way enabling the linkage between unstructured and structured data and knowledge collections. A major challenge in this endeavor lies in the dynamics of the digital contents about the world, with new entities emerging all the time. In this paper, we propose a crowdsourced framework for NERD, specifically addressing the challenge of emerging entities in social media. Our approach combines NERD techniques with the detection of entity alias names and with co-reference resolution in texts. We propose a linking-game based crowdsourcing system for this combined task, and we report on experimental insights with this approach and on lessons learned.

Keywords: Named Entity Recognition and Disambiguation, Crowdsourcing

1 Introduction

Knowledge bases, linked data, and other semantic web assets are flourishing [9, 23, 12, 26] and contribute to improved search, analytics, and recommendation services. These assets contain many billions of facts about many millions of entities like people, places, companies, music bands, songs, diseases, drugs, proteins, etc. Additional value is created by *entity-level links* that span collections, via RDF triples with the owl:sameAs predicate [9, 11]. This way, different collections complement each other. For example, while one data source knows everything about the musicians of a song, another one contains data about the sales of the song's album, and yet another one knows about the use of the song in movies or cover versions by other artists. Jointly, this allows analyzing a musician's influence on the entertainment industry.

Structured data will hardly ever be complete, as there is always some detail not captured in RDF triples and the world is rapidly evolving anyway. Therefore, it is crucial to establish also entity-level links between unstructured sources like news articles or social media and the web of linked open data. Manually creating microdata embedded in HTML pages is one approach, but this will still leave many gaps. To fill these gaps, largely automated methods are needed, discovering names of entities in text, tables, or lists of surface web contents and mapping them to entities in linked-data collections. As names can have many different meanings, this entails the need for *Named Entity Recognition and Disambiguation (NERD)*.

Fully automatic NERD is inherently difficult and may also be computationally expensive (see, e.g., [18, 15, 10, 22, 13, 2]). NERD performs very well for prominent entities in high-quality texts like news articles, but they degrade in precision and recall when dealing with long-tail entities and difficult inputs like social media. Since advanced methods utilize machine learning or extensive statistics for semantic relatedness measures among entities, the availability of labeled training data is usually a big bottleneck. This is one of issues where crowdsourcing [4] can help, in order to improve NERD quality.

Even if we had perfect NERD methods, the cross-linkage between unstructured web contents and semantic data collections would still have big gaps. The reason is the *dynamics* of the world: new entities come into existence (e.g., songs, hurricanes, scandals) and unnoted entities suddenly gain importance (e.g., Edward Snowden, Adele two years ago). When facing such *emerging entities*, we cannot map them to a knowledge base (yet) as there are no RDF triples about them. However, we can capture their mentions under different names and try to gather equivalence classes of text phrases that refer to the same entity. This is known as the task of *coreference resolution (CR)* (see, e.g., [8, 20, 21, 24]). For example, we should discover the mentions "Edward Snowden", "NSA agent Snowden", and "the Prism whistleblower" and infer that they denote the same emerging entity, while also inferring that "actress Snowden" and "CEO Snowden" are separate entities.

CR methods can also help to increase the recall of NERD for known entities, by capturing more surface phrases (e.g., [17, 19]). For example, the German football team FC Bayern Munich may be known and detectable as "Bayern Munich", "FC Bayern', or as "Germany's most successful football club", but the additional name "triple winner" makes sense only since end of May 2013 (when the team won three major championships). If, for a given text, we infer that "triple winner" and "UEFA champion 2013" are the same entity, we can map more text mentions onto entities, thus improving NERD recall at high precision. Systematically gathering alias names for entities is the problem of *alias detection (AD)*. It has been studied in the literature, harnessing href anchor texts, click logs, and other assets (see, e.g., [14, 25]). However, doing this for emerging entities that are not yet registered in a knowledge base is a largely unexplored task.

The goal of this paper is to address the above problems in creating semantic markup for entities. Our approach is unique in that we address the three problems NERD, CR, and AD in a joint manner. Our methodology is *crowdsourcing*: asking people to annotate text snippet (e.g., tweet). While this approach may seem straightforward, it does come with technical challenges. First, we need to cast the problem into simple user interactions so that laymen can contribute with little effort. Second, we need to cope with highly varying quality of user contributions. Third, we need to optimize the benefit/cost ratio, by making judicious choices about which text snippets are shown to which people.

This paper presents a first cut on these problems, including experimental studies. The benefit of our crowdsourcing architecture is twofold: i) we create semantic markup in the form of co-reference between mentions, which can be directly used as input for methods that connect the web of unstructured contents with the web of linked data at the entity level, and ii) we lay the foundation to use this annotated contents to improve automated methods for NERD, CR, and AD. In the future, by continuously running a low-cost crowdsourcing process on news or social media, we can periodically re-train and re-configure automated methods and adjust them to the dynamics of web contents.

2 Related Work

NERD methods [18, 15, 10, 22, 13, 2] aim to identify entity mentions in natural-language text and weakly structured web contents like HTML tables and lists, and link the mentions to entities registered in a knowledge base or linked data source. Coreference resolution (CR) identifies mentions in text that refer to the same entity [8, 20, 21, 24], but without mapping them onto data or knowledge bases. Note that these tasks are fairly different from database-oriented task of entity resolution, aka. entity matching or record linkage [7], which is solely focused on structured records (with known schema) as input.

Crowdsourcing [4, 16, 5, 3] harnesses human input for tasks that are inherently difficult for computers, such as image tagging or language understanding. Approaches along these lines come in two major families: i) explicit crowdsourcing with HITs (human intelligence tasks) assigned to paid workers on platforms like Amazon Mechanical Turk (`www.mturk.com`) or CrowdFlower (`crowdflower.com`), and ii) implicit crowdsourcing where the task is piggybacked on human-computer interactions or in the form of a game.

Crowdsourcing was used for the problem of entity resolution [27] on structured database records. Recall that this task is quite different from our problem of NERD and CR over text snippets. This work also compares a list-wise with a pair-wise style user interface. In contrast, we aim to compare user behavior under different user interfaces (i.e., pair-wise and linking-game based interface).

3 Overview of Methodology

We have developed a framework for combining NERD, CR, and AD. Figure 1 gives a pictorial overview. The emphasis in this paper is on crowdsourcing the task of CR, in the form of a linking game, and harness the user feedback obtain this way for improving AD and NERD.

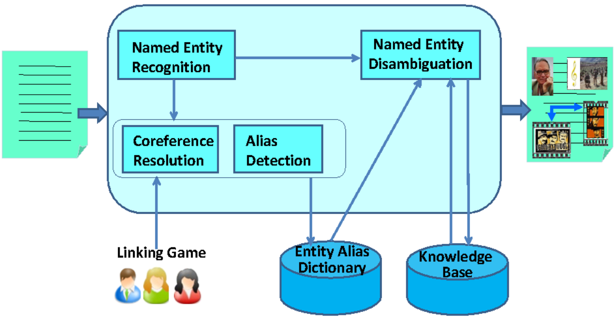


Fig. 1. Framework for NERD, CR, and AD.

In the following we briefly characterize the functionality of each component, and explain the dataflow between components.

Named Entity Recognition (NER). The input text is processed to discover *mentions* of named entities, that is, surface phrases that are likely to denote individual entities (as opposed to common noun phrases). Our implementation currently uses the Stanford NER Tagger [6] for this purpose (a trained CRF).

Crowdsourced Coreference Resolution (CR). All mentions in the same input text are highlighted and presented to human players, using a game-like interface. The participating users are asked to connect mentions that refer to the same entity. This way we obtain equivalence classes of mentions. Note that this does not perform any disambiguation yet: we still do not know which entity an equivalence class of mentions refers to, and in the case of newly emerging entities may not have the proper entity registered in our knowledge base anyway.

Alias Detection (AD). The CR step has the benefit of providing us with alias names for the same entity. Some of these names may already be present in our dictionary of entity aliases (e.g., “the US president’s wife” for Michelle Obama), but others are new discoveries (e.g., “the First Lady of the White House”). If we can later, in the NED step, map the entire equivalence class of coreferences to an entity, we can easily add the new aliases to the dictionary. This way, we improve the AD task and increase the coverage of our dictionary.

Named Entity Disambiguation (NED). Finally, we attempt to map all mentions to canonicalized entities registered in a knowledge base. We use the YAGO knowledge base for this purpose (`http://yago-knowledge.org`), but can easily switch to other choices like DBpedia or Freebase. The actual NED computation is based on the AIDA method [10] and its open-source software (`https://github.com/yago-naga/aida`). AIDA combines context-similarity measures with coherence measures for the entities chosen for different mentions. We have further extended AIDA to become aware of the coreference equivalence classes obtained in the CR step. This extension is presented in Section 5.

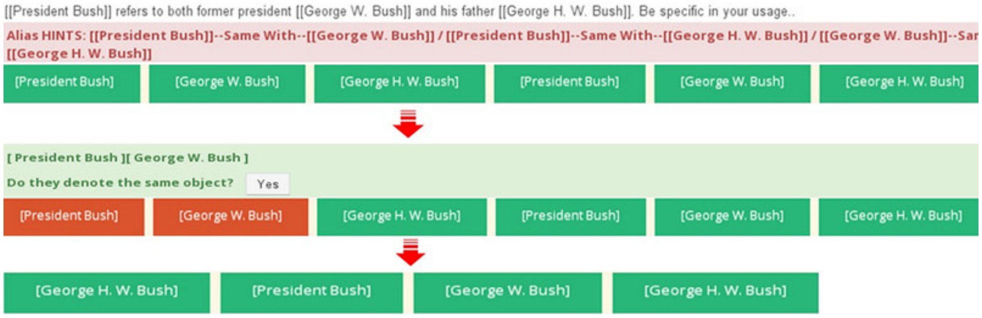
4 Crowdsourced Coreference Resolution

4.1 Mention Linking Game

We created a crowdsourcing interface that allows humans to highlight coreferenced mentions in a text snippet in a light-weight manner. To minimize the burden on humans and as an additional incentive, we developed a game-like interface inspired by the “Linking Game”[1], in which players earn points by finding identical icons in an image. This in turn is reminiscent of the well-known Concentration Game, also known as Memory, just with all cards already open.

Figure 2 shows a sequence of three screenshots of our mention linking game. Players are asked to mark up all co-referent mentions for a given set of mentions highlighted in the text. The user receives hints about which mentions may possibly be equivalent, using simple heuristics for automated CR. All mentions are then presented as green blocks for markup by the user. When the user selects blocks, they are turned red. Once the user clicks on “Yes” to confirm that they are coreferences, these blocks are removed from the

[1] `http://www.appszoom.com/android_games/sports_games/cute-puppys-link-game_bsddz.html`



[[President Bush]] refers to both former president [[George W. Bush]] and his father [[George H. W. Bush]]. Be specific in your usage..

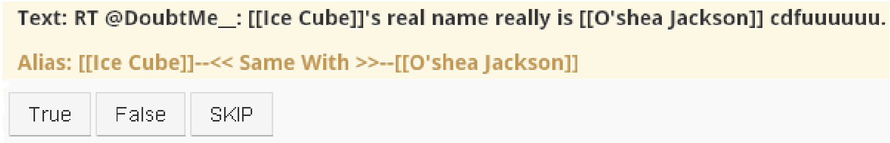
Alias HINTS: [[President Bush]]--Same With--[[George W. Bush]] / [[President Bush]]--Same With--[[George H. W. Bush]] / [[George W. Bush]]--Sar [[George H. W. Bush]]

[President Bush][George W. Bush]

Do they denote the same object?

Fig. 2. Linking-Game Interface

user's view. When players are very certain about one selection, they can select the same equivalent mention pair multiple times. This gives us an implicit way of estimating the confidence of a user's input.



Text: RT @DoubtMe__: [[Ice Cube]]'s real name really is [[O'shea Jackson]] cdfuuuuuu.

Alias: [[Ice Cube]]--<< Same With >>--[[O'shea Jackson]]

Fig. 3. Pair-wise User Interface

To compare the effectiveness of the linking-game based interface against more traditional crowdsourcing interfaces, we also designed game UI for judging each pair of mentions separately, as shown in Figure 3. A pair of mentions is presented, and the player has to make one of the three choices: Yes, No, or Skip.

4.2 Quality Control

For assessing the quality of the players, we prepared a set of gold-standard texts for which we identified the correct equivalence classes of mentions. These gold-standard texts are occasionally presented as linking-game tasks, and a user's performance on these is a first-cut estimate for the confidence in the user's markup.

4.3 Feedback for Automated Coreference Resolution

High-confidence annotations obtained from the game are chosen as the crowdsourced results of CR. These results are directly used to enhance named entity disambiguation, as described in the following Section. Additionally, high quality annotations can be used as training data. The samples will help to better learn feature weights, where features could be alias matching, abbreviation/acronym matching, string similarity, position relative to the two mentions of interest, part-of-speech tags, etc. Details of this enhancement and its performance are beyond the scope of this paper.

5 Combining NERD, CR, and AD

We used the AIDA tool [10] as a basis for our crowdsourcing-enhanced NERD method. AIDA works in four steps. First, it uses the Stanford NER Tagger to identify mentions in the input text. Second, it generates candidate entities by looking up the surface names in the dictionary and retrieving the associated entities from the knowledge base. Third, it builds a graph that connects mention nodes with candidate entity nodes by edges that are weighted with context-similarity scores, and connects pairs of candidate entity nodes by edges that are weighted with semantic coherence scores. Fourth and last, AIDA runs an algorithm for computing a dense subgraph whose entity nodes yield the desired disambiguation. Figure 4, upper part, shows an example graph with these two types of edges. The graph contains a third kind of edges, connecting pairs of mention nodes. These are actually added by our crowdsourced-CR process, as explained in Section 5.

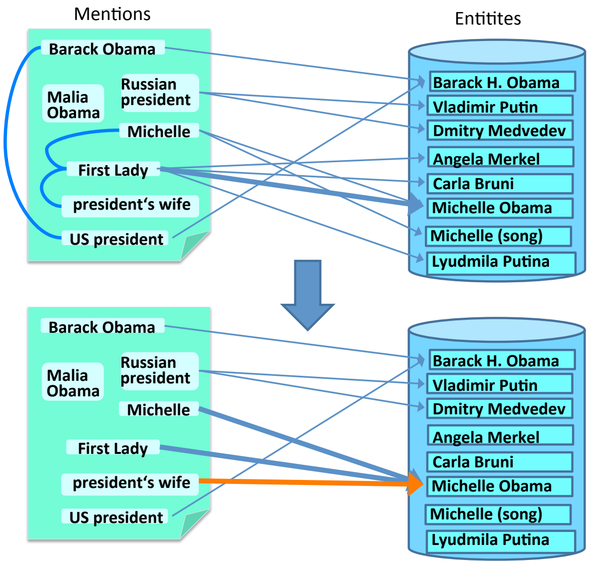


Fig. 4. Example graph for combined NED and CR

In the example, “Michelle” is a highly ambiguous mention, which is difficult to map to the proper entity. Here, the crowdsourced CR yields valuable input by linking this mention with the other two mentions “president’s wife” and “First Lady”, thus easing the tasks of NED. Note that some of the mentions marked up in the NER step may not be in the dictionary; so usually no candidate entities would be generated for a mention such as “president’s wife”. By the CR markup from the crowdsourcing phase, we can transfer the candidate entities from other mentions, “First Lady” and “Michelle”, to this newly recognized phrase. We actually choose the entity that has the highest weight among all the candidates in the same CR equivalence class for all the mentions. Finally note that one mention in the example text, “Malia Obama” is not linkable to the knowledge base at all, as there there is no suitable entity there.

Our enhancements of AIDA work by extending the mapping graph. For every set of mentions, $m_1, m_2, \ldots, m_k$, that were combined into one equivalence class by the crowdsourced CR, we proceed as follows:

- Case 1: All of $m_1, m_2, \ldots, m_k$ have matches in the dictionary. In this case, we generate all respective candidate entities, by lookups in the knowledge base, and then choose the highest weighted entity among all candidate sets, retaining only this entity for all mentions in the CR equivalence class.
- Case 2: The set $M = \{m_1, m_2, \ldots, m_k\}$ contains some mentions that do not have any matches in the alias dictionary, say subset $N \subset M$. In this case, we determine the entity for the potentially linkable mentions, subset $L = M - N$, according to Case 1 and then add it to all mentions in N.

  In addition, we insert the mentions in N as new alias names for the retained candidate entities into the alias dictionary, thus enhancing the AD component of our framework.
- Case 3: None of the mentions in $M = \{m_1, m_2, \ldots, m_k\}$ has any match in the dictionary, so they are all non-linkable. In this case, we drop these mentions from the NED graph. However, we do insert this set of mentions into the alias dictionary as alias names for an unknown entity. This can pay off later, for a new input text, if that text has a CR equivalence class that includes both a name associated with a known entity and an alias from M. This way, we potentially improved both AD and NERD in the long run.

The lower part of Figure 4 shows the graph that results from these steps. After these graph-enhancement steps, all mention-mention edges are removed. The resulting graph can be directly fed into the AIDA tool for the actual NED computation.

6 Experimental Results

6.1 Experimental Setup

In our preliminary studies reported here, we focus on two types of entities from tweets: *persons* and *locations*. We used lists of 50 US states and 50 celebrities, from the prior work of [1] (http://www.iba.t.u-tokyo.ac.jp/~danushka/data/aliasdata.zip). Each entity comes with a small number of alias names. For example, Michael Jordan (the basketball player) has alias names “Air Jordan”, “His Airness”, and “MJ”, and Whoopi Goldberg is also known as “Da Whoop” and “Caryn Elaine Johnson”.

We further extended this dataset in three ways. First, we included additional persons (all US presidents) and locations (a set of large cities around the world) as concerned entities. This led to a total of 93 person entities and 150 location entities. Second, we gathered tweets from Twitter (twitter.com) by generating queries with the entity names and their alias names. Third, we added tweets from the UK election 2009. We selected 140 tweets for crowdsourcing experiment, and 100 tweets for NED evaluation. The number of mentions are counted by using a liberal NER method, combining the Stanford Tagger [6] and a dictionary-based matcher for entity names and aliases. Our complete experimental data is available at http://www.mpi-inf.mpg.de/yago-naga/aida/download/iswc-crowdsem2013.zip.

6.2 CR Performance

A total of 14 university students participated in our crowdsourcing experiment, 7 playing the linking game and 7 using the pair-wise UI. For evaluation, we manually annotated 140 tweets. We aggregated the human contributions for the same tweet by weighted voting, where weights reflect the confidence in a user (which in turn is based on how well the user performed for the occasional gold-standard inputs, see Section 4.2). We compared the two crowdsourcing settings against a fully automated heuristic algorithm for CR, based on the following simple rules:

- When two mentions exactly match aliases for the same entity in our dictionary, the algorithm connects them into a CR equivalence class.
- When two mentions have high string similarity above a threshold, the algorithm connects them.
- When the text between two mentions contains a strong pattern such as “also known as”, “called”, “referred to”, etc., the algorithm connects them.

The results in terms of precision, recall, and F1 scores are shown in Table 1. We observe that the Linking-Game-based crowdsourcing clearly outperformed the pair-wise annotator UI. This is due to the vastly increased number of decisions necessary for pair-wise annotators, which increases the risk of making mistakes. The game-based crowdsourced CR also won against the rule-based algorithm by a large margin, in terms of F1 scores. However, the experiment also revealed trade-offs: the automatic algorithm did much better in terms of recall, but was much inferior to the crowdsourced CR in terms of precision.

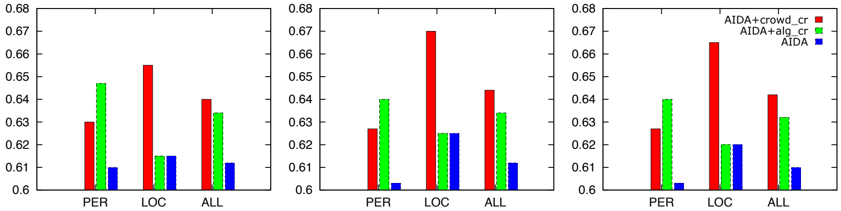
Mention Type	**Linking Game**			**Pair-wise UI**			**Algorithm**		
	Prec.	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**
person	**0.85**	0.70	0.77	0.52	0.80	0.63	0.53	**1.0**	0.69
location	**0.98**	0.81	0.88	0.61	0.54	0.57	0.58	**1.0**	0.73
overall	**0.92**	0.76	0.83	0.56	0.67	0.60	0.55	**0.99**	0.71

Table 1. Linking-Game vs. Pair-wise-UI vs. Algorithm Results for CR

6.3 NED Performance

We manually mapped the mentions in 100 tweets onto proper entities for as ground-truth for experiments on NED performance. We compared three methods: the standard AIDA method, our enhancement using crowdsourced CR annotations (see Section 5), an analogous enhancement of AIDA by CR annotations obtain from the rule-based heuristic algorithm (see CR experiments above). The results are shown in Figure 5.

The results clearly show that the combined CR+NED approach (AIDA+alg_cr) achieves much better performance than the state-of-the-art NED method (AIDA) alone. When comparing the influence of crowdsourced CR vs. algorithmic CR, we see mixed results: none of these two methods dominates the other. However, in terms of overall F1 score across all mentions, the crowdsourcing-enhanced method (AIDA+crowd_cr) is the overall winner.



(a) Precision (b) Recall (c) F1 Measure

Fig. 5. NED Performance Comparison

7 Lessons Learned

This paper presented a new approach to combining NED (Named Entity Disambiguation), CR (Coreference Resolution), and AD (Alias Detection) with crowdsourcing-based CR. Our experiments are a first proof of concept that this directions is worthwhile being pursued further at larger scale. The Linking-Game-based interface turned out to yield better results than a more traditional annotator UI. This is an encouragement towards intensifying and extending this game-based approach.

As for the overall improvement that CR contributes to NED performance, our experiments, albeit still small-scaled, clearly indicate that CR annotations are very beneficial for NED. Moreover, they also contribute to maintaining the alias name dictionary and thus handling emerging entities. As for the crowdsourced vs. algorithmic CR (see Table 1), the situation is less clear, though. The crowdsourcing approach has both higher precision and recall, however, it still has weaknesses when text snippets are very demanding. For example, consider the tweet: “The Rich are Running from California. The once Golden State is trying to bail itself out by going after the rich.” Realizing that “California” and “Golden State” denote the same entity was beyond what our crowdsourcing users could do, so our approach failed on this sample. Co-occurrence statistics for mentions, mined from Web and text corpora, could overcome this weakness. This calls for a new hybrid between crowdsourced and algorithmic methods.

8 Acknowledgements

This work is supported by the 7th Framework IST programme of the European Union through the focused research project(STREP) on Longitudinal Analytics of Web Archive data (LAWA) under contract no. 258105.

References

1. D. Bollegala, T. Honma, Y. Matsuo, and M. Ishizuka. Automatically extracting personal name aliases from the web. In *Proceedings of the 6th international conference on Advances in Natural Language Processing*, GoTAL ’08, pages 77–88, 2008.
2. M. Cornolti, P. Ferragina, and M. Ciaramita. A framework for benchmarking entity-annotation systems. In *WWW*, pages 249–260, 2013.
3. G. Demartini, D. E. Difallah, and P. Cudré-Mauroux. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *WWW*, pages 469–478, 2012.

4. A. Doan, R. Ramakrishnan, and A. Y. Halevy. Crowdsourcing systems on the world-wide web. *Commun. ACM*, 54(4):86–96, 2011.
5. T. Finin, W. Murnane, A. Karandikar, N. Keller, J. Martineau, and M. Dredze. Annotating named entities in twitter data with crowdsourcing. In *CSLDAMT*, pages 80–88, 2010.
6. J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*, ACL '05, pages 363–370, 2005.
7. L. Getoor and A. Machanavajjhala. Entity resolution: theory, practice & open challenges. *Proc. VLDB Endow.*, 5(12):2018–2019, Aug. 2012.
8. A. Haghighi and D. Klein. Simple coreference resolution with rich syntactic and semantic features. In *EMNLP*, pages 1152–1161, 2009.
9. T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers, 2011.
10. J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust disambiguation of named entities in text. In *Proceedings of EMNLP*, EMNLP '11, pages 782–792, 2011.
11. A. Hogan, J. Umbrich, A. Harth, R. Cyganiak, A. Polleres, and S. Decker. An empirical survey of linked data conformance. *J. Web Sem.*, 14:14–44, 2012.
12. E. H. Hovy, R. Navigli, and S. P. Ponzetto. *Special Issue on Artificial Intelligence, Wikipedia and Semi-Structured Resources*, volume 194. 2013.
13. R. Isele and C. Bizer. Learning expressive linkage rules using genetic programming. *PVLDB*, 5(11), 2012.
14. L. Jiang, J. Wang, P. Luo, N. An, and M. Wang. Towards alias detection without string similarity: an active learning based approach. In *SIGIR*, pages 1155–1156, 2012.
15. S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of wikipedia entities in web text. In *KDD*, pages 457–466, 2009.
16. E. Law and L. von Ahn. *Human Computation*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2011.
17. T. Lin, Mausam, and O. Etzioni. No noun phrase left behind: Detecting and typing unlinkable entities. In *EMNLP-CoNLL*, pages 893–903, 2012.
18. D. N. Milne and I. H. Witten. Learning to link with wikipedia. In *CIKM*, pages 509–518, 2008.
19. N. Nakashole, T. Tylenda, and G. Weikum. Fine-grained semantic typing of emerging entities. In *ACL*, page to appear, 2013.
20. A. Rahman and V. Ng. Coreference resolution with world knowledge. In *ACL*, pages 814–824, 2011.
21. L.-A. Ratinov and D. Roth. Learning-based multi-sieve co-reference resolution with knowledge. In *EMNLP-CoNLL*, pages 1234–1244, 2012.
22. L.-A. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and global algorithms for disambiguation to wikipedia. In *ACL*, pages 1375–1384, 2011.
23. N. Shadbolt, K. O'Hara, T. Berners-Lee, N. Gibbins, H. Glaser, W. Hall, and m. c. schraefel. Linked open government data: Lessons from data.gov.uk. *IEEE Intelligent Systems*, 27(3):16–24, 2012.
24. S. Singh, A. Subramanya, F. C. N. Pereira, and A. McCallum. Large-scale cross-document coreference using distributed inference and hierarchical models. In *ACL*, pages 793–803, 2011.
25. V. I. Spitkovsky and A. X. Chang. A cross-lingual dictionary for english wikipedia concepts. In *LREC*, pages 3168–3175, 2012.
26. F. M. Suchanek and G. Weikum. Knowledge harvesting in the big-data era. In *SIGMOD Conference*, pages 933–938, 2013.
27. J. Wang, T. Kraska, M. J. Franklin, and J. Feng. Crowder: crowdsourcing entity resolution. *Proc. VLDB Endowment*, 5(11):1483–1494, July 2012.

Frame Semantics Annotation Made Easy with DBpedia

Marco Fossati, Sara Tonelli, and Claudio Giuliano
{fossati, satonelli, giuliano}@fbk.eu

Fondazione Bruno Kessler - via Sommarive, 18 - 38123 Trento, Italy

Abstract. Crowdsourcing techniques applied to natural language processing have recently experienced a steady growth and represent a cheap and fast, albeit valid, solution to create benchmarks and training data. Nevertheless, some particularly complex tasks such as semantic role annotation have been rarely conducted in a crowdsourcing environment, due to their intrinsic difficulty. In this paper, we present a novel approach to accomplish this task by leveraging information automatically extracted from DBpedia. We show that replacing role definitions, typically meant for expert annotators, with a list of DBpedia types, makes the identification and assignment of role labels more intuitive also for non-expert workers. Results prove that such strategy improves on the standard annotation workflow, both in terms of accuracy and of time consumption.

Keywords: Natural Language Processing, Frame Semantics, Entity Linking, DBpedia, Crowdsourcing, Task Modeling

1 Introduction

Frame semantics [6] is one of the theories that originate from the long strand of linguistic research in artificial intelligence. A *frame* can be informally defined as an event triggered by some term in a text and embedding a set of participants. For instance, the sentence `Goofy has murdered Mickey Mouse` evokes the Killing frame (triggered by `murdered`) together with the Killer and Victim participants (respectively `Goofy` and `Mickey Mouse`). Such theory has led to the creation of FrameNet [2], namely an English lexical database containing manually annotated textual examples of frame usage.

Annotating frame information is a complex task, usually modeled in two steps: given a sentence, annotators are first asked to choose the frame activated by a predicate (or *lexical unit, LU*, e.g. `murdered` in the example above evoking Killing). Second, they assign the semantic roles (or *frame elements, FEs*) that describe the participants involved in the chosen frame. In this work, we focus on the second step, namely FEs recognition.

Currently, FrameNet development follows a strict protocol for data annotation and quality control. The entire procedure is known to be both time-consuming and costly, thus representing a burden for the extension of the resource [1]. Furthermore, deep linguistic knowledge is needed to tackle this annotation task, and the resource developed so far would not have come to light without the contribution of skilled linguists and lexicographers. On one hand, the task complexity depends on the inherently complex theory behind frame semantics, with a repository of thousands of roles available for the assignment. On the other hand, these roles are defined for expert annotators, and their descriptions are often obscure to common readers. We report three examples below:

- Support: Support is a fact that lends epistemic support to a claim, or that provides a reason for a course of action. Typically it is expressed as an External Argument. (EVIDENCE frame)
- Protagonist: A person or self-directed entity whose actions may potentially change the mind of the Cognizer (INFLUENCE_OF_EVENT_ON_COGNIZER frame)
- Locale: A stable bounded area. It is typically the designation of the nouns of Locale-derived frames. (LOCALE_BY_USE frame)

Since we aim at investigating whether such activity can be cast to a crowd of non-expert contributors, we need to reduce its complexity by intervening on the FE descriptions. In particular, we want to assess to what extent more information on the role semantics coming from external knowledge sources such as DBpedia[1] can improve non-expert annotators' performance. We leverage the CrowdFlower platform,[2] which serves as a bridge to a plethora of crowdsourcing services, the most popular being Amazon's Mechanical Turk (AMT).[3]

We claim that providing annotators with information on the semantic types typically associated with FEs will enable faster and cheaper annotations, while maintaining an equivalent accuracy. The additional information is extracted in a completely automatic way, and the workflow we present can be potentially applied to any crowdsourced annotation task in which semantic typing is relevant.

2 Related Work

The construction of annotation datasets for natural language processing tasks via non-expert contributors has been approached in different ways, the most prominent being *games with a purpose* (GWAP) and *micro-tasks*. While the former technique leverages fun as the motivation for attracting participants, the latter mainly relies on a monetary reward. The effects of such factors on a contributor's behavior have been analyzed in the motivation theory literature, but are beyond the scope of this paper. The reader may refer to [10] for an overview focusing on AMT.

Games with a Purpose. Verbosity [17] was one of the first attempts in gathering annotations with a GWAP. Phrase Detectives [5,4] was meant to harvest a corpus with coreference resolution annotations. The game included a validation mode, where participants could assess the quality of previous contributions. A data unit, namely a resolved coreference for a given entity, is judged complete only if the agreement is unanimous. Disagreement between experts and

[1] `http://dbpedia.org`

[2] `https://crowdflower.com`

[3] `https://www.mturk.com`

the crowd appeared to be a potential indicator of ambiguous input data. Indeed, it has been shown that in most cases disagreement did not represent a poor annotation, but rather a valid alternative.

Micro-tasks. Design and evaluation guidelines for five natural language micro-tasks are described in [15]. Similarly to our approach, the authors compared crowdsourced annotations with expert ones for quality estimation. Moreover, they used the collected annotations as training sets for machine learning classifiers and measured their performance. However, they explicitly chose a set of tasks that could be easily understood by non-expert contributors. Similarly, [13] built a multilingual textual entailment dataset for statistical machine translation by developing an annotation pipeline to decompose the annotators' task into a sequence of activities. Finally, [8] exploited Google AdWords, a tool for web advertising, to measure message persuasiveness while avoiding subjects being aware of the experiments and being biased by external rewards.

Semantic Role Annotation. Manual annotation of semantic roles has been recently addressed via crowdsourcing in [9] and [7]. Furthermore, [1] highlighted the crucial role of recruiting people from the crowd in order to bypass the need for linguistics expert annotations. Uniformly to our contribution, the task described in [9] was modeled in a multiple-choice answers fashion. Nevertheless, the focus is narrowed to the frame discrimination task, namely selecting the correct frame evoked by a given LU. Such task is comparable to the word sense disambiguation one as per [15], although the difficulty seems augmented, due to lower inter-annotator agreement values. The authors experienced issues that are related to our work with respect to the quality check mechanism in CrowdFlower, as well as the complexity of the frame names and definitions. Outsourcing the task to the CrowdFlower platform has two major drawbacks: (a) the proprietary nature of the aggregated inter-annotator agreement value provided in the response data, and (b) the need to manually simplify FE definitions that generated high disagreement. In this respect, our previous work [7] was the first attempt to address item (b) by manually simplifying the way FEs are described. In this work, we further investigate this aspect by exploiting automatically extracted links to DBPedia.

3 Annotation Workflow

Our goal is to determine if crowdsourced annotation of semantic roles can be improved by providing non-expert annotators with information from DBpedia on the roles they are supposed to label. Specifically, instead of displaying the lexicographic definition for each possible role to be labeled, annotators are shown a set of semantic types associated with each role coming from FrameNet. Based on this, annotators should better recognize such roles in an unseen sentence. Evaluation is performed by comparing this annotation framework with a baseline, where standard FE definitions substitute DBpedia information.

Before performing the annotation task, we need to leverage the list of semantic types that best characterizes each FE in a frame. We extract these statistics by connecting the FrameNet database 1.5 [14] to DBpedia, after isolating a set of sentences to be used as test data (cf. Section 4). The workflow to prepare the input for the crowdsourced task is based on the following steps.

Linking to Wikipedia. For each annotated sentence in the FrameNet database, we first link each textual span labeled as FE to a Wikipedia page W. We employ *The Wiki Machine*, a kernel-based linking system (details on the implementation are reported in [16]), which was trained on the Wikipedia dump of March 2010.[4] Since FEs can be expressed by both common nouns and real-world entities, we needed a linking system that satisfactorily processes both nominal types. A comparison with the state-of-the-art system *Wikipedia Miner* [12] on the ACE05-WIKI dataset [3] showed that The Wiki Machine achieved a suitable performance on both types (.76 F1 on real-world entities and .63 on common nouns), while Wikipedia Miner had a poorer performance on the second noun type (respectively .76 and .40 F1). These results were also confirmed in a more recent evaluation [11], in which The Wiki Machine achieved the highest F1 compared with an ensemble of academic and commercial systems, such as *DBpedia Spotlight*, *Zemanta*, *Open Calais*, *Alchemy API*, and *Ontos*.

The system applies an ‘all word’ linking strategy, in that it tries to connect each word (or multiword) in a given sentence to a Wikipedia page. In case a linked textual span (partially) matches a string corresponding to a FE, we assume that one possible sense of FE is represented in Wikipedia through W. The Wiki Machine also assigns a confidence score to each linked term. This confidence is higher in case the words occurring in the same context of the linked term show high similarity, because the system considers that the linking is likely to be more accurate.

We illustrate in Figure 1 the Wikipedia pages (and confidence score) that the Wiki Machine system associates with the sentence `Sardar Patel was assisting Gandhiji in the Salt Satyagraha with great wisdom`, an example sentence for the ASSISTANCE frame originally annotated with four FEs, namely Helper, Benefited_party, Goal and Manner. Since Wikipedia is a repository of concepts, which are usually expressed by nouns, we are able to link only nominal fillers.

Linking to DBpedia. In order to obtain the semantic types that are typical for each FE, linking to Wikipedia is not enough. In fact, too many different pages would be connected to a FE, making it difficult to generalize over the Wikipedia pages (i.e. concepts). This emerges also from the example above, where the pages linked to `Sardar Patel`, `Gandhjii` and `Salt Satyagraha` do not provide information on the typical fillers of Helper, Benefited_party and Goal respectively. One possible option could be to resort to Wikipedia categories, which however are not homogenous enough to allow for a consistent extraction of FE semantic types.

We tackle this problem by using Wikipedia pages as a bridge to DBpedia. In fact, Wikipedia page URLs directly map to DBpedia resource URIs. Hence, for each linked FE, we query DBpedia for `rdf:type` objects. In this way, we are able to compute statistics on the most frequent semantic types associated with

[4] `http://download.wikimedia.org/enwiki/20100312`

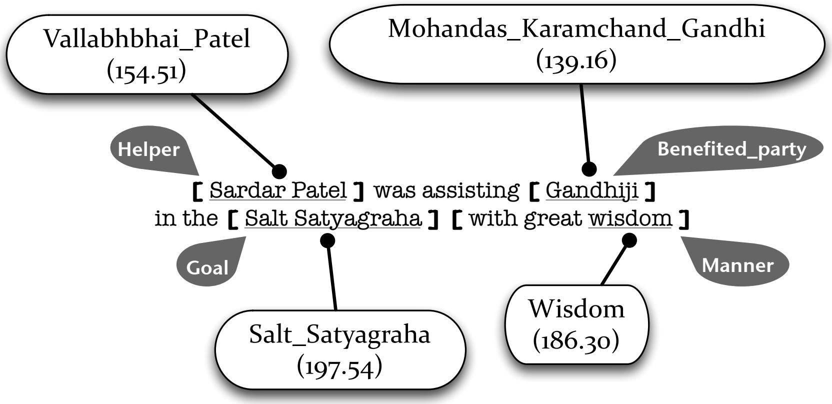

Vallabhbhai_Patel
(154.51)
Mohandas_Karamchand_Gandhi
(139.16)
Helper
Benefited_party
[Sardar Patel] was assisting [Gandhiji]
in the [Salt Satyagraha] [with great wisdom]
Goal
Manner
Wisdom
(186.30)
Salt_Satyagraha
(197.54)

Fig. 1: Linking example with confidence score

a given FE from a given frame. For instance, the FE Victim from the KILLING frame has a top DBpedia type *Animal* with a frequency of 38. We aim at investigating whether such top-occurring types represent both valid generalizations and simplifications of a standard FE definition, and may thus substitute it. At the end of this pre-processing step, we create a repository where, for each FE, a set of DBpedia types is listed and ranked by frequency.

Posting the Annotation Task on CrowdFlower. We finally set up a crowdsourced experiment where, in each test sentence, annotators have to choose the most appropriate FE given the most frequent DBpedia types (proper task) or the standard FE definition (baseline). Details are reported in the following section.

4 Experiments

We first provide an overview of critical aspects underpinning a generic crowdsourced experiment. Subsequently, we describe the anatomy and the modeling of the tasks we outsourced to the CrowdFlower platform. Input data, full results, interface code and screenshots are available at `http://db.tt/iogsU7RI` .

Golden Data. Quality control of the collected judgements is a key factor for the success of the experiments. The essential drawback of crowdsourcing services relies on the cheating risk. Workers are generally paid a few cents for tasks which may only need a single click to be completed. Hence, it is highly probable to collect data coming from random choices that can heavily pollute the results. The issue is resolved by adding *gold* units, namely data for which the requester already knows the answer. If a worker misses too many gold answers within a given threshold, he or she will be flagged as untrusted and his or her judgments will be automatically discarded.

Worker Switching Effect. Depending on their accuracy in providing answers to gold units, workers may switch from a trusted to an untrusted status and vice versa. In practice, a worker submits his or her responses via a web page. Each page contains one gold unit and a variable number of regular units that can be set by the requester during the calibration phase. If a worker becomes untrusted, the platform collects another judgment to fill the gap. If a worker moves back to the trusted status, his or her previous contribution is added to the results as free extra judgments. Such phenomenon typically occurs when the complexity of gold units is high enough to induce low agreement in workers' answers. Thus, the requester is constrained to review gold units and to eventually forgive workers who missed them. This has not been a blocking issue in our experiments, since we assessed a relatively low average percentage of missed judgments for gold units, namely 28%.

Cost Calibration. The total cost of a crowdsourced task is naturally bound to a data unit. This represents an issue in our experiments, as the number of questions per unit (i.e. a sentence) varies according to the number of frames and FEs evoked by the LU contained in a sentence. Therefore, we need to use the average number of questions per sentence as a multiplier to a constant cost per sentence. We set the payment per working page to 3 $ cents and the number of sentences per page to 3. Since most of the sentences in our annotation task have 3 FEs, the average cost per FE results in 0.325 $ cent (see Table 2 below).

Pre-processing of FrameNet Data for DBpedia Types Extraction. Table 1 provides some statistics of the processed FrameNet data that were leveraged to extract DBpedia types (cf. Section 3). More specifically:

1. From the FrameNet 1.5 database, the Wiki Machine managed to link 77% of the total number of FE instances. Hence, unlinked data is skipped for the next step.
2. DBpedia provided type information for 42% of the total number of linked FE instances. Types occurring once are ignored, as they reflect the content of a single sentence and are likely to convey misleading suggestions. The too generic `owl#Thing` type is filtered as well.

Table 1: FrameNet data processing details

Workflow step	**FE instances**
Raw FrameNet	148440
Linking to Wikipedia	114242
DBpedia types extraction	47732

Test Data Preparation. Before linking the FrameNet database to DBpedia, we isolate a subset to be used as test data. From 500 randomly chosen sentences, we select those in which the number of FEs per frame is between 3 and 4.

This small dataset serves as input for our experiments. Table 2 details the final settings. We hand-pick six sentences and for each of them we mark one question as gold for quality check. Almost all sentences contain three FEs with few exceptions (cf. the average value in Table 2). We extract the five most frequent DBpedia types from the statistics and assign them to the corresponding FEs in our input. Since not all FEs have exactly five associated types (cf. the average value in Table 2), we provide workers with variable suggestion sets. Finally, we ensure all workers are native English speakers.

Table 2: Experimental settings

Sentences	43
Gold	6
Frames	24
Lexical Units	41
Average FEs per sentence	3.07
Average cost per FE ($ cents)	.325
Average DBpedia types per FE	4.66
Workers nationality	United States

Modeling. Data units are delivered to workers via a web interface. Our task is illustrated in Figure 2 and is presented as follows:

(a) Workers are invited to read a sentence and to focus on the bolded word appearing as a title above the sentence (e.g. `taste` in the screenshot).

(b) A question concerning each FE is then shown together with a set of answers corresponding to the sentence chunks that may express the given FE. For instance, in Figure 2, the question `Which is the Perceiver Passive?` is coupled with multiple choices taken from the given sentence.

(c) For each question, a suggestion box displays the top types retrieved from DBpedia and connected to the given FE (cf. Section 3 for details). This should help annotators in choosing the text chunk that better fits the given FE.

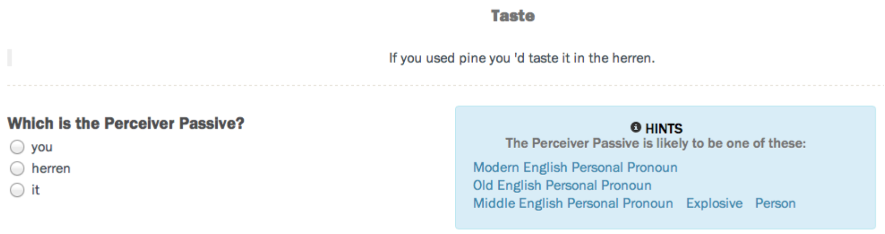
(d) Finally, workers match each question with the proper text chunk.

On the other hand, the baseline differs from our strategy in that (i) it does not display the suggestion box and (ii) questions are replaced with the FE definition extracted from FrameNet. For instance, in Figure 2, the question about the Perceiver Passive would be replaced with `This FE is the being who has a perceptual experience, not necessarily on purpose`. The baseline is more compliant with the standard approach adopted to annotate FEs in the FrameNet project.

5 Results

Our main purpose is to evaluate the validity of the proposed approach against the conventional FrameNet annotation procedure. We leverage expert-annotated sentences and are thus able to directly measure workers' accuracy. Specifically, we compute 2 values:

- *Majority vote.* An answer is considered correct only if the majority of judgments are correct.



Taste

If you used pine you 'd taste it in the herren.

Which is the Perceiver Passive?

○ you
○ herren
○ it

HINTS

The Perceiver Passive is likely to be one of these:

Modern English Personal Pronoun
Old English Personal Pronoun
Middle English Personal Pronoun Explosive Person

Fig. 2: Worker interface unit screenshot

- *Absolute.* The total number of correct judgments divided by the total number of collected judgments.

The results of our experiments are detailed in Table 3. The number of untrusted judgments may be considered as a shallow indicator of the overall task complexity. In fact, we tried to maximize objectivity and simplicity when choosing gold units. Moreover, the input dataset (and gold units as well) is identical in both experiments. Therefore, we can infer that the number of workers who missed gold is directly influenced by the question model, which is the only variable parameter. We compute the execution time as the interval between the first and the last judged unit.

Table 3: Overview of the experimental results

Measure	Baseline	DBpedia
Majority vote accuracy	.763	.803
Absolute accuracy	.646	.720
Untrusted judgments	90	82
Time (minutes)	160	106

Our approach outperformed the baseline both in terms of accuracy and time. While majority vote accuracy values differ slightly, absolute accuracy clearly favors our strategy. Such measure can be seen as a further indicator of the task complexity. A higher score implies a higher number of correct judgments, which may designate a better inter-worker agreement, thus a more straightforward task. This claim is not only supported by the moderate decrease of untrusted judgments, but also by the dramatic reduction of the execution time. Consequently, the results we obtained demonstrate that entity linking techniques combined with DBpedia types simplify FEs annotation.

6 Discussion and Conclusions

In this work, we present a novel approach to annotate frame elements in a crowdsourcing environment using information extracted from DBpedia. The task is simplified for non-expert annotators by replacing FE definitions, usually meant for linguistic experts, with semantic types obtained from DBpedia. This is accomplished without manual simplification, in a completely automatic fashion.

Results prove that such method improves on the standard annotation workflow, both in terms of accuracy and of time consumption. Although the interconnection between FEs and DBpedia is semantically not perfect, extracting frequency statistics from the whole FrameNet database and considering only the most occurring types from DBpedia make the procedure quite robust to wrong links.

Possible issues may arise when two or more frame elements in the same frame share the same semantic type. For instance, the *Goal* and *Place* FEs in the ARRIVING frame are both likely to be filled by elements describing a location. We also expect that our approach is less accurate with FEs that can be filled both by nouns and by verbs, for instance the *Activity* FE in the ACTIVITY_FINISH frame. In such cases, information extracted from DBpedia would probably be inconsistent. Besides, DBpedia statistics are reliable when several annotated sentences are available for a frame, while they may be misleading if extracted from few instances. We plan to investigate these issues and to explore possible solutions to cope with data sparseness.

Additional future work will involve the following aspects:

- Evaluation of an ad-hoc strategy for the extraction of semantic types, namely providing workers with suggestions by matching information that are dynamically derived from each given sentence with DBpedia types.
- Clustering of similar semantic types with respect to the meaning they convey and to the frequency, e.g. `Place` and `Location_Underspecified`.

Finally, the overall effectiveness of our approach depends both on the performance of the entity linking system and on the coverage of the knowledge base. Hence, long term research will focus on enhancing The Wiki Machine precision and recall, and extending DBpedia type coverage.

References

1. Baker, C.F.: FrameNet, current collaborations and future goals. Language Resources and Evaluation pp. 1–18 (2012)
2. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The Berkeley FrameNet Project. In: Proceedings of the 17th international conference on Computational linguistics-Volume 1. pp. 86–90. Association for Computational Linguistics (1998)
3. Bentivogli, L., Forner, P., Giuliano, C., Marchetti, A., Pianta, E., Tymoshenko, K.: Extending English ACE 2005 Corpus Annotation with Ground-truth Links to Wikipedia. In: 23rd International Conference on Computational Linguistics. pp. 19–26 (2010)
4. Chamberlain, J., Kruschwitz, U., Poesio, M.: Constructing an anaphorically annotated corpus with non-experts: Assessing the quality of collaborative annotations. In: Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources. pp. 57–62. Association for Computational Linguistics (2009)
5. Chamberlain, J., Poesio, M., Kruschwitz, U.: Phrase detectives: A web-based collaborative annotation game. Proceedings of I-Semantics, Graz (2008)
6. Fillmore, C.: Frame semantics. Linguistics in the morning calm pp. 111–137 (1982)

7. Fossati, M., Giuliano, C., Tonelli, S.: Outsourcing FrameNet to the Crowd. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. pp. 742–747. Sofia, Bulgaria (August 2013), `http://www.aclweb.org/anthology/P13-2130`
8. Guerini, M., Strapparava, C., Stock, O.: Ecological Evaluation of Persuasive Messages Using Google AdWords. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics. ACL2012 (July 2012)
9. Hong, J., Baker, C.F.: How Good is the Crowd at "real" WSD? In: Proceedings of the 5th Linguistic Annotation Workshop. pp. 30–37 (2011)
10. Kaufmann, N., Schulze, T., Veit, D.: More than fun and money. Worker motivation in crowdsourcing – A study on Mechanical Turk. In: Proceedings of the Seventeenth Americas Conference on Information Systems, Detroit, MI (2011)
11. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: Dbpedia spotlight: shedding light on the web of documents. In: Proceedings of the 7th International Conference on Semantic Systems. pp. 1–8. I-Semantics '11, ACM, New York, NY, USA (2011)
12. Milne, D., Witten, I.H.: Learning to link with Wikipedia. In: CIKM '08: Proceedings of the 17th ACM conference on Information and knowledge management. pp. 509–518. ACM, NY, USA (2008)
13. Negri, M., Bentivogli, L., Mehdad, Y., Giampiccolo, D., Marchetti, A.: Divide and conquer: crowdsourcing the creation of cross-lingual textual entailment corpora. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 670–679. EMNLP '11, Association for Computational Linguistics, Stroudsburg, PA, USA (2011)
14. Ruppenhofer, J., Ellsworth, M., Petruck, M.R., Johnson, C.R., Scheffczyk, J.: FrameNet II: Extended Theory and Practice. Available at http://framenet.icsi.berkeley.edu/book/book.html (2006)
15. Snow, R., O'Connor, B., Jurafsky, D., Ng, A.Y.: Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 254–263. Association for Computational Linguistics (2008)
16. Tonelli, S., Giuliano, C., Tymoshenko, K.: Wikipedia-based WSD for multilingual frame annotation. Artificial Intelligence 194, 203–221 (2013)
17. Von Ahn, L., Kedia, M., Blum, M.: Verbosity: a game for collecting common-sense facts. In: Proceedings of the SIGCHI conference on Human Factors in computing systems. pp. 75–78. ACM (2006)

Developing Crowdsourced Ontology Engineering Tasks: An iterative process

Jonathan M. Mortensen, Mark A. Musen, Natalya F. Noy

Stanford Center for Biomedical Informatics Research
Stanford University, Stanford CA 94305, USA

Abstract. It is increasingly evident that the realization of the Semantic Web will require not only computation, but also *human* contribution. Crowdsourcing is becoming a popular method to inject this human element. Researchers have shown how crowdsourcing can contribute to managing semantic data. One particular area that requires significant human curation is ontology engineering. Verifying large and complex ontologies is a challenging and expensive task. Recently, we have demonstrated that online, crowdsourced workers can assist with ontology verification. Specifically, in our work we sought to answer the following driving questions: (1) Is crowdsourcing ontology verification feasible? (2) What is the optimal formulation of the verification task? (3) How does this crowdsourcing method perform in an application? In this work, we summarize the experiments we developed to answer these questions and the results of each experiment. Through iterative task design, we found that workers could reach an accuracy of 88% when verifying SNOMED CT. We then discuss the practical knowledge we have gained from these experiments. This work shows the potential that crowdsourcing has to offer other ontology engineering tasks and provides a template one might follow when developing such methods.

1 Background

Research communities have begun using crowdsourcing to assist with managing the massive scale of data we have today. Indeed, certain tasks are better solved by humans than by computers. In the life sciences, Zooniverse, a platform wherein citizen scientists contribute to large scale studies, asks users to perform tasks such as classifying millions of galaxies or identifying cancer cells in an image [8]. In related work, Von Ahn and colleagues developed games with a purpose, a type of crowdsourcing where participants play a game, and as a result help complete some meaningful task. For example, in Fold.it, gamers assist with folding a protein, a computationally challenging task [4]. Further demonstrating the power of the crowd, Bernstein et al. developed a system that uses the crowd to quickly and accurately edit documents [1]. With crowdsourcing's popularity rising, many developer resources are now available, such as Amazon's Mechanical Turk, Crowdflower, oDesk, Houdini, etc. Finally, as evidenced by this workshop, CrowdSem, the Semantic Web community is beginning to leverage crowdsourcing. Systems such as CrowdMap, OntoGame, and ZenCrowd demonstrate how crowdsourcing can contribute to the Semantic Web [11, 2, 10]. Crowdsourcing enables the completion of tasks at a massive scale that cannot be done computationally or by a single human.

One area amenable to crowdsourcing is ontology engineering. Ontologies are complex, large, and traditionally require human curation, making their development an ideal candidate task for crowdsourcing. In our previous work, we developed a method for crowdsourcing ontology verification. Specifically, we sought to answer the following driving questions:

(1) Is crowdsourcing ontology verification feasible?
(2) What is the optimal formulation of the verification task?
(3) How does this crowdsourcing method perform in an application?

In this work, we briefly highlight each of the experiments we developed to answer our questions, and, with their results in mind, then discuss how one might approach designing crowdsourcing tasks for the Semantic Web. In previous work, we have published papers that explore each driving question in depth. The main contribution of this work is a unified framework that presents all of the experiments. This framework will enable us to reflect on current work and to ask new questions for crowdsourcing ontology engineering.

2 Ontology Verification Task

We have begun to reduce portions of ontology engineering into microtasks that can be solved through crowdsourcing. We devised a microtask method of ontology verification based on a study by Evermann and Fang [3] wherein participants answer computer-generated questions about ontology axioms . A participant verifies if a sentence about two concepts that are in a parent-child relationship is correct or incorrect. For example, the following question is a hierarchy-verification microtask for an ontology that contains classes Heart and Organ:

Is every Heart an Organ?

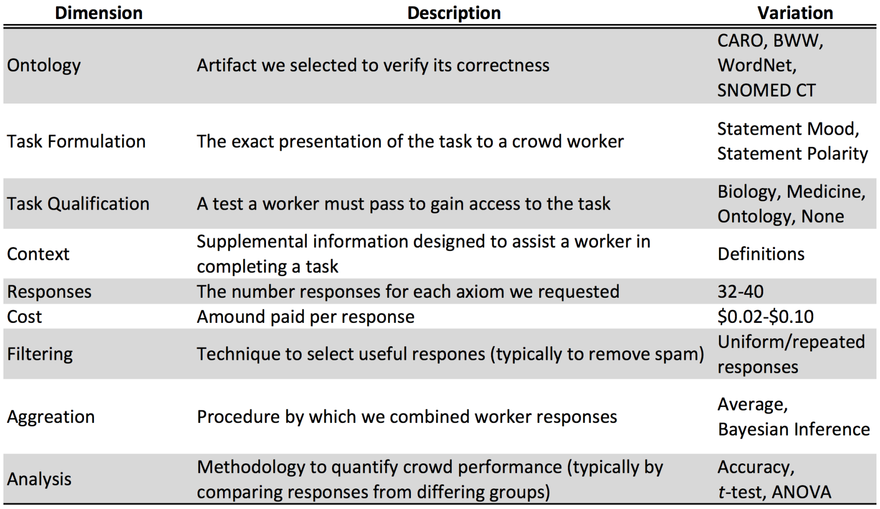
A worker then answers the question with a binary response of “Yes” or “No.”

This task is particularly useful in verifying ontologies because the class hierarchy is the main type of relationship found in many ontologies. For example, in 296 public ontologies in the BioPortal repository, 54% of these ontologies contained only SubClassOf relationships between classes. In 68% of ontologies, the SubClassOf relationships accounted for more than 80% of all relationships. Thus, verifying how well the class hierarchy corresponds to the domain will enable the verification of a large fraction of the relations in ontologies.

3 Protocol & Experimental Design

We developed various experiments that use the hierarchy verification task to answer our driving questions. Generally, each of these experiments follows the same basic procedure. First, we selected the ontology and axioms to verify. Next, we created a hierarchy-verification task formatted as HTML from these entities and submitted the task to Amazon Mechanical Turk. Finally, we obtained worker responses, removed spam, and compared the remaining responses to a gold standard using some analysis metric. Thus, in each experiment we used a standard set of basic components outlined in Table 1. Typically, we manipulated one of these components in each experiment. Figure 1 presents an example task as it appears to a worker on Amazon Mechanical Turk.

Table 1. Dimensions in which our crowdsourcing experiments vary.



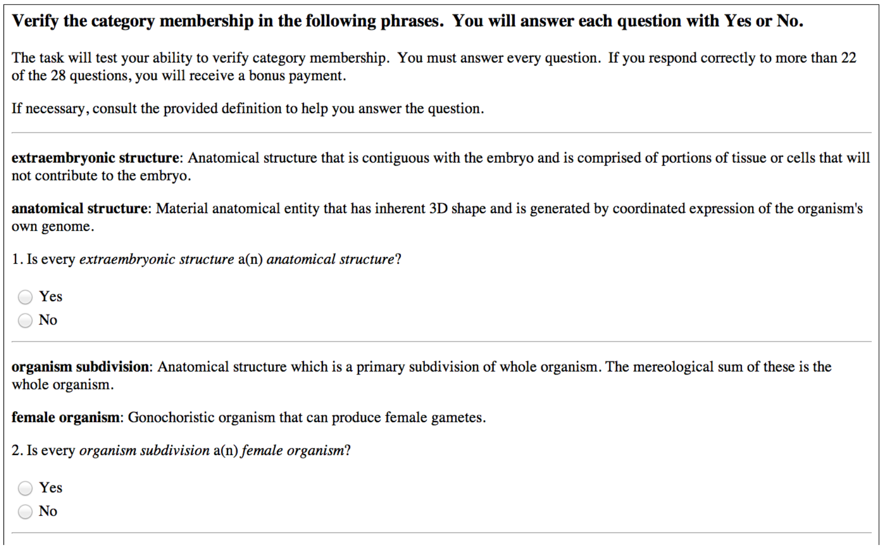
Dimension	Description	Variation
Ontology	Artifact we selected to verify its correctness	CARO, BWW, WordNet, SNOMED CT
Task Formulation	The exact presentation of the task to a crowd worker	Statement Mood, Statement Polarity
Task Qualification	A test a worker must pass to gain access to the task	Biology, Medicine, Ontology, None
Context	Supplemental information designed to assist a worker in completing a task	Definitions
Responses	The number responses for each axiom we requested	32-40
Cost	Amound paid per response	\$0.02-\$0.10
Filtering	Technique to select useful respones (typically to remove spam)	Uniform/repeated responses
Aggreation	Procedure by which we combined worker responses	Average, Bayesian Inference
Analysis	Methodology to quantify crowd performance (typically by comparing responses from differing groups)	Accuracy, *t*-test, ANOVA

4 Experiments

To answer the driving questions, we performed a series of experiments using the basic protocol. We began with the most basic question about feasibility of the method. Having shown that, we tested various parameters in order to optimize the method. Finally, we used the optimal method in verifying SNOMED CT. Table 2 summarizes these experiments and their parameters. In the following, we describe the specifics of each experiment and our conclusions for each.

4.1 Is crowdsourcing ontology verification feasible? [7]

In this first driving question, we wished to understand if it were possible for Amazon Mechanical Turk workers (turkers) to perform on par with other groups also performing the hierarchy-verification task.



Verify the category membership in the following phrases. You will answer each question with Yes or No.

The task will test your ability to verify category membership. You must answer every question. If you respond correctly to more than 22 of the 28 questions, you will receive a bonus payment.

If necessary, consult the provided definition to help you answer the question.

extraembryonic structure: Anatomical structure that is contiguous with the embryo and is comprised of portions of tissue or cells that will not contribute to the embryo.

anatomical structure: Material anatomical entity that has inherent 3D shape and is generated by coordinated expression of the organism's own genome.

1. Is every *extraembryonic structure* a(n) *anatomical structure*?

○ Yes
○ No

organism subdivision: Anatomical structure which is a primary subdivision of whole organism. The mereological sum of these is the whole organism.

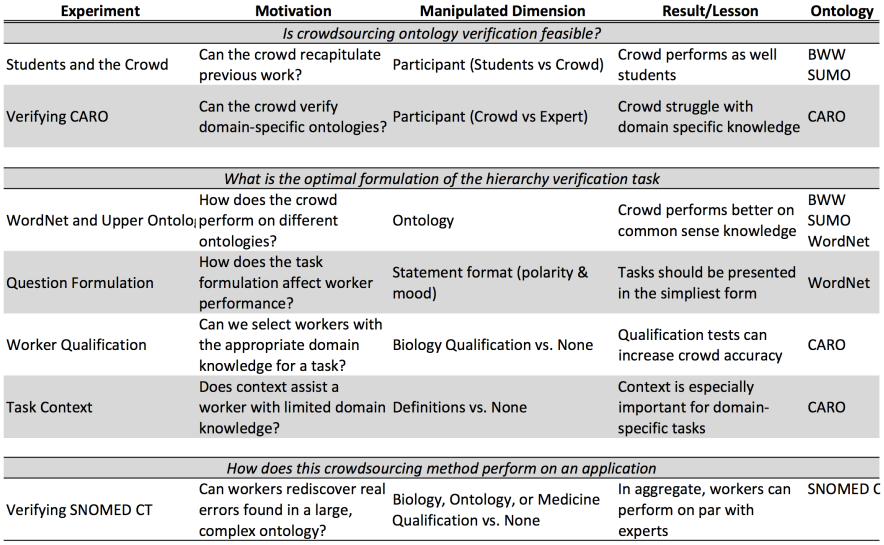
female organism: Gonochoristic organism that can produce female gametes.

2. Is every *organism subdivision* a(n) *female organism*?

○ Yes
○ No

Figure 1. Example task that a Mechanical Turk worker sees in a browser. In this example, workers are provided concept definitions and a Subsumption relation from the Common Anatomy Reference Ontology to verify.

Table 2. Experiments we performed in developing a method to crowdsource ontology verification.



Experiment	Motivation	Manipulated Dimension	Result/Lesson	Ontology
		Is crowdsourcing ontology verification feasible?		
Students and the Crowd	Can the crowd recapitulate previous work?	Participant (Students vs Crowd)	Crowd performs as well students	BWW SUMO
Verifying CARO	Can the crowd verify domain-specific ontologies?	Participant (Crowd vs Expert)	Crowd struggle with domain specific knowledge	CARO
		What is the optimal formulation of the hierarchy verification task		
WordNet and Upper Ontolo	How does the crowd perform on different ontologies?	Ontology	Crowd performs better on common sense knowledge	BWW SUMO WordNet
Question Formulation	How does the task formulation affect worker performance?	Statement format (polarity & mood)	Tasks should be presented in the simpliest form	WordNet
Worker Qualification	Can we select workers with the appropriate domain knowledge for a task?	Biology Qualification vs. None	Qualification tests can increase crowd accuracy	CARO
Task Context	Does context assist a worker with limited domain knowledge?	Definitions vs. None	Context is especially important for domain-specific tasks	CARO
		How does this crowdsourcing method perform on an application		
Verifying SNOMED CT	Can workers rediscover real errors found in a large, complex ontology?	Biology, Ontology, or Medicine Qualification vs. None	In aggregate, workers can perform on par with experts	SNOMED C

Experiment 1: Students and the Crowd

Methods We determined whether turkers could recapitulate results from a study by Evermann and Fang [3]. In this study, after completing a training session, 32 students performed the hierarchy-verification task with 28 statements from the Bunge-Wand-Weber ontology (BWW) and 28 statements from Suggested Upper Merged Ontology (SUMO), where half of the statements were true, and half false in each. As an incentive to perform well, students were offered a reward for the best performance.

Knowing the results of that experiment, we asked turkers to verify the same statements. As in the initial study, we required turkers to complete a 12 question training qualification test. We asked for 32 turkers to answer each 28 question set and paid $0.10/set. Furthermore, we offered a bonus for good performance. After turkers completed the tasks, we removed spam responses from workers who responded with more than 23 identical answers. Finally, we compared the performance of the students with that of the turkers using a paired t-test

Results In both experiments, the average accuracy of student was 3–4% higher than the accuracy of the turkers. However, the difference was not statistically significant.

Conclusion Turkers recapitulated previous hierarchy-verification results and performed on par with students in the hierarchy-verification task.

Experiment 2: Verifying the Common Anatomy Reference Ontology (CARO)

Methods Verifying a domain ontology was the second component in showing the feasibility of our verification method. For this verification task, we used CARO, a well curated biomedical ontology. We selected 14 parent-child relations from the ontology as correct relations. Like with WordNet, we paired children with parents that were not in the same hierarchy to simulate incorrect relations. We then asked workers to verify these relations following the earlier experimental setup. In this situation, we had no qualification test. As a comparison, we asked experts on the obo-anatomy and National Center for Biomedical Ontology mailing lists to perform the same verification. Finally, we measured worker and expert performance, and compared the groups using a t-test.

Results With the proper task design of context and qualifications (addressed later), turkers performed 5% less accurately than experts, but there was not a statistically significant difference.

Conclusions Workers performed nearly as as well as experts in verifying a domain ontology. These results are quite encouraging. In addition, the results of this experiment led us to hypothesize that worker performance significantly depends on the task formulation. We address this next.

4.2 What is the optimal formulation of the hierarchy verification task? [5]

With the feasibility of crowdsourcing ontology verification established, we focused on formulating the task in an optimal fashion. There were four main parameters that we hypothesized would affect the method's performance: Ontology Type (i.e., the domain of the ontology being verified), Question Formulation (i.e., How should we ask a worker to verify a relationship?), Worker Qualification (i.e., How does the accuracy of a worker vary based on certain qualification?), and Context (i.e., What information should be provide to assist a worker in answering the question?).

Experiment 3: WordNet and Upper Ontologies

Methods Having shown that turkers perform similarly to students and domain experts, we then analyzed how turker performance varied based on ontology selection. To do so, we compared worker performance in verifying BWW and SUMO to verifying WordNet. We created a set of WordNet statements to verify by extracting parent-child relationships in WordNet and also generating incorrect statements from incorrectly paired concepts (i.e. pairing concepts in parent-child relationships that are not actually hierarchically related). We then asked workers to verify the 28 WordNet, SUMO and BWW statements following the same setup as the first experiment (including the training qualification), paying workers $0.10/set, giving a bonus, and removing spam.

Results Echoing the first experiment, workers performed only slightly better than random on BWW and SUMO, respectively. However, workers had an average accuracy of 89% verifying WordNet statements. There was a clear difference between worker performance on upper ontologies and WordNet.

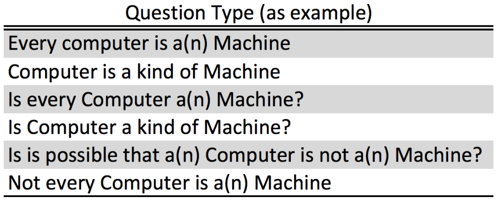
Conclusion While workers struggle with verifying conceptually difficult relationships, such as those contained in upper level ontologies, they perform reasonably well in tasks related to common-sense knowledge.

Experiment 4: Question Formulation

Methods We repeated the task of verifying 28 WordNet statements but varied the polarity and mood of the verification question we ask the workers. In this case, we did not require qualifications as with the earlier experiments. Table 3 shows the 6 different question styles through example.

Results Worker performance varied from 77% on negatively phrased statements to 91% with the positive, indicative mood (i.e., a True/False statement asserting the relationship). In addition, workers responded faster with positively phrased questions.

Table 3. Example question types we presented to users on Mechanical Turk.



Question Type (as example)
Every computer is a(n) Machine
Computer is a kind of Machine
Is every Computer a(n) Machine?
Is Computer a kind of Machine?
Is is possible that a(n) Computer is not a(n) Machine?
Not every Computer is a(n) Machine

Conclusion Generally for crowdsourcing, one should create tasks in the most cognitively simple format as possible. In this situation, asking the verification as simply as possible (i.e., Dog is a kind of Mammal. True or False?)

Experiment 5: Worker Qualification

Methods Having determined the optimal method to ask the verification question, we theorized that workers who could pass a domain-specific qualification test would perform better than a random worker on tasks related to that domain. We developed a 12 question high-school level biology qualification test. For turkers to access our tasks, they would have to pass this test. We assume that the ability to pass this test serves as a reasonable predictor of biology domain knowledge. We asked workers to complete the CARO verification (Experiment 3), but required them to first pass the qualification task, answering at least 50% of it correctly.

Results With qualifications, turkers improved their accuracy to 67% (from random without qualifications) when verifying CARO.

Conclusion When crowdsourcing, some method to select experts in the domain of the task is necessary to achieve reasonable performance. However, such low accuracy was not satisfying to the authors.

Experiment 6: Task Context

Methods With the increases in performance with proper question formulation and qualification requirements, we next proposed that concept definitions would assist workers in verifying a relation. In this experiment, we used CARO because the ontology has a complete set of definitions. We repeated Experiment 3, with qualifications and simply stated verification questions, varying whether users and experts were shown definitions.

Results With definitions, workers performed with an average accuracy of 82%. Experts performed with an average accuracy of 86%. So, when providing workers and experts with definitions, there was no statistically significant difference.

Conclusion In crowdsourcing, context is essential, especially for non-domain experts. While workers might not have very specific domain knowledge, with proper context or training, they can complete the task. This experiment revealed that in some situations, a properly designed microtask can indeed provide results on par with experts.

4.3 How does this crowdsourcing method perform on an application? [6]

The previous experiments were all synthetic – turkers only found errors that we introduced. With the optimal task formulation in hand, we shifted our focus to a true ontology verification task of verifying a portion of SNOMED CT. We selected SNOMED CT because it is a heavily studied, large and complex ontology, making it an ideal candidate for our work.

Experiment 7: Verifying SNOMED CT

Methods In 2011, Alan Rector and colleagues identified entailed SubClass axioms that were in error [9]. In our final experiment, we evaluated whether our method could recapitulate their findings. To do so, we asked workers to perform the hierarchy verification task on these 7 relations along with 7 related relations we already knew were correct. We used the optimal task formulation we determined in earlier experiments and provided definitions from the Unified Medical Language System. In addition, we posted the task with 4 different qualification tests: biology, medicine, ontology, and none. To note, instead of asking workers to complete the task of verifying all 14 relations in one go, as with earlier experiments, we instead broke up the task into smaller units, creating one task per axiom and paid unqualified workers and qualified workers $0.02 and $0.03 per verification, respectively. We then compared worker's average performance to their aggregate performance (i.e., when we combined all workers responses to one final response through majority voting [6]).

Results The aggregate worker response was 88% accurate in differentiating correct versus incorrect SNOMED CT relations. On average, any single worker performed 17% less accurately than the aggregate response. Furthermore, there was no significant difference in performance for tasks with differing qualification tests.

Conclusion Individually, workers did not perform well in identifying errors in SNOMED CT. However, as a group, they perform quite well. The stark difference between average worker performance and aggregate performance reinforces the fact that the power of the crowd lies in their combined response, not any worker alone.

5 Discussion

Each of the experiments we performed highlighted various lessons we learned in developing a method for crowdsourcing ontology verification. A few lessons are particularly useful for the Semantic Web community. First, many of our experiments focused on changing small components of the task. Even so, through this process we greatly improved crowd worker performance. It is clear that each task will be unique, but in most cases, extensive controlled trials will assist in identifying the best way to crowdsource a task. Following this strategy, we verified a complex ontology with relatively high accuracy. In addition, our current results only serve as a baseline – through additional iteration, we expect the increases in accuracy to continue.

Second, using the refined tasks, we showed that crowd workers, in aggregate, can perform on par with experts on domain specific tasks when provided with simple tasks and the proper context. The addition of context was the single biggest factor at improving performance. In the Semantic Web, a trove of structured data are available, all of which may provide such needed context (and maybe other elements, such as qualification tests). For example, when using the crowd to classify instance-level data, the class hierarchy, definitions, or other instance examples may all assist the crowd in their task.

Our results suggest that crowdsourcing might serve as method to improve other ontology engineering tasks such as typing instances, adding definitions, creating ontology mappings and even ontology development itself. In fact, Sarasua and colleagues used crowdsourcing to improve automated ontology mapping methods [10]. ZenCrowd follows a similar paradigm, using crowdsourcing to improve machine extracted links [2]. Indeed, crowdsourcing can serve as a human curated step in ontology engineering that acts in concert with automated methods (e.g., terminology induction supplemented with the crowd).

5.1 Future Work

The results thus far serve only as a baseline for crowdsourcing an ontology engineering task. We plan to focus research on other elements in the crowdsourcing pipeline, include entity selection (e.g., selecting the axioms for verification that will most likely be in error), generating context (e.g., how can we use the crowd to also supply context for workers downstream), and optimizing performance (e.g., developing aggregation strategies that maximize worker performance while minimizing task cost). We will also consider different incentives models including reputation or altruism, like the successful Zooniverse platform [8]. Finally, we will investigate how to integrate this method into a true ontology engineering workflow with the Protege ontology development platform.

6 Conclusion

Crowdsourcing is now another tool for the Semantic Web researcher and developer. In this work, we described various experiments we performed to refine a methodology to crowdsource ontology verification. In summary, we arrived at a highly accurate method through iterative, controlled development of the crowdsourcing task. In doing so, we gained valuable knowledge about method design for crowdsourcing. For example, providing task context is key to enabling accurate crowd workers. Finally, our results suggest that crowdsourcing can indeed contribute to ontology engineering.

Acknowledgements

This work is supported in part by Grant GM086587 from the National Institute of General Medical Sciences and by The National Center for Biomedical Ontology, supported by grant HG004028 from the National Human Genome Research Institute and the National Institutes of Health Common Fund. JMM is supported by National Library of Medicine Informatics Training Grant LM007033.

References

1. Bernstein, M., Little, G., Miller, R., Hartmann, B., Ackerman, M., Karger, D., Crowell, D., Panovich, K.: Soylent: a word processor with a crowd inside. In: The 23d annual ACM symposium on user interface software and technology. pp. 313–322. ACM (2010)
2. Demartini, G., Difallah, D.E., Cudré-Mauroux, P.: ZenCrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In: 21st World Wide Web Conference WWW2012. pp. 469–478. Lyon, France (2012)
3. Evermann, J., Fang, J.: Evaluating ontologies: Towards a cognitive measure of quality. Information Systems 35, 391-403 (2010)
4. Khatib, F., DiMaio, F., Cooper, S., Kazmierczyk, M., Gilski, M., Krzywda, S., Zabranska, H., Pichova, I., Thompson, J., Popović, Z., Jaskolski, M., Baker, D.: Crystal structure of a monomeric retroviral protease solved by protein folding game players. Nat Struct Mol Biol 18(10), 1175–1177 (10 2011)
5. Mortensen, J.M., Alexander, P.R., Musen, M.A., Noy, N.F.: Crowdsourcing Ontology Verification. In: International Conference on Biomedical Ontologies (2013).
6. Mortensen, J.M., Musen, M.A., Noy, N.F.: Crowdsourcing the Verification of Relationships in Biomedical Ontologies. In: AMIA Annual Symposium. Accepted (2013)
7. Noy, N.F., Mortensen, J.M., Alexander, P.R., Musen, M.A.: Mechanical Turk as an Ontology Engineer? Using Microtasks as a Component of an Ontology Engineering Workflow. In: Web Science (2013)
8. Raddick, M.J., Bracey, G., Gay, P.L., Lintott, C.J., Cardamone, C., Murray, P., Schawinski, K., Szalay, A.S., Vandenberg, J.: Galaxy Zoo: Motivations of Citizen Scientists p. 41 (Mar 2013
9. Rector, A.L., Brandt, S., Schneider, T.: Getting the foot out of the pelvis: modeling problems affecting use of SNOMED CT hierarchies in practical applications. Journal of the American Medical Informatics Association 18(4), 432–440 (Apr 2011)
10. Sarasua, C., Simperl, E., Noy, N.F.: CrowdMAP: Crowdsourcing Ontology Alignment with Microtasks. In: 11th International Semantic Web Conference (ISWC). Springer, Boston, MA (2012)
11. Siorpaes, K., Hepp, M.: Games with a Purpose for the Semantic Web. IEEE Intelligent Systems 23(3), 50–60 (May 2008)

Information Reputation

Peter Davis[1] and Salman Haq[2]

[1] Distinguished Engineer,
Neustar, 21575 Ridgetop Circle, Sterling, Virginia, USA
peter.davis@neustar.biz,
[2] Research Architect,
Neustar, 21575 Ridgetop Circle, Sterling, VA, USA
salman.haq@neustar.biz

Abstract. In this paper we describe the design of a *reputation framework* for an information management system under active development. The integration of a reputation framework with an IMS is a novel combination that can produce a distinctly more effective business intelligence tool.

1 Introduction

Neustar is a data analytics and intelligence services company that operates several large database systems. To efficiently manage these numerous, disparate systems, we are developing an Information Management System (IMS) that maps technical data models using a standard set of ontologies. The IMS is an online community for employees where they can share, classify and discover metadata about various Neustar *data source*s. Its main purpose is to assist users in achieving two main objectives: *a*) reducing costs by utilizing existing information and *b*) increasing revenues by creating new information [3].

With these objectives in mind, users must have the ability to make value judgments about *data source*s relative to one another. Such *data source*s may number in the hundreds and the *datum* contained therein may number in the tens of thousands. The majority of these entities will lack significant value for data science, and those that are valuable will risk being lost in a deluge of information. Therefore it is imperative that the system establish a bias towards meaningful *datum* by highlighting *interestingness*. A well-crafted reputation framework can excel at doing exactly this.

2 Terminology

Glossary

claim One or more assertions made of a *datum*.
data source A computer system that stores data such as a database or file system.

data steward A individual or group of individuals holding domain-specific knowledge of an information system.

datum An instance of metadata mapped to an atomic data field. This includes, for example, columns in a relational database or entities defined in an XML schema.

interestingness A scalar value indicating the suitability for inclusion in further analysis.

reputation A qualitative measure that informs a value judgment about a *datum* or user.

Acronyms

IMS Information Management System.

3 Framework Description

The framework is comprised of several *reputation* models, each of which computes one or more scores for a resource type. A fixed set of *claim*s serve as inputs to each model which assigns numerical values to them and passes them through a series of mathematical filter *processes*. Models are distinguished by their input selection, process configuration, and output scores. The IMS utilizes a fixed ontology to define *claim*s that include appropriate business and technical classifications for data within the subject systems. The essential *claim*s of the *datum* model are summarized in Table 1. The IMS also incorporates techniques to simplify crowd sourcing the classification of *datum* by *data steward*s. However, we have concluded from early usage, that a simple classification process is insufficient. Classifications can be subjective, and classification sparseness results in under utilization of the system. As a consequence, methods to encourage accurate and complete classification will be implemented to enrich the overall efficacy of the system.

Table 1. Essential *Claim*s for the *Datum* reputation model

Name	**Description**
classified	*Data steward* classified *datum* from the business domain ontology
described	*Data steward* entered a description
discussed	User participated in a discussion topic about *datum*
emailed	User emailed the link to the *datum* page to another user
flagged	User informed the *data steward* about insufficient or inaccurate details
watched	User will be notified of future updates by other users
wanted	User requested access to the *datum* from the *data steward*

4 Interestingness Reputation Model for Datum

In the IMS, *datum* is an atomic unit of data. Its' classification results in queriable metadata, and can relate to a column in a relational database or an element, attribute or phrase in a document. At the time of writing, the system had *over 7,000 fields from merely four* data source*s*. Even at this watermark, the task of finding *interesting datum* is impractical for any user community. As more *data source*s are imported into the system, this task will be become impossible even if the datum population grows sub-linearly. Therefore it is imperative that the system is capable of identifying and highlighting interesting datum to facilitate user objectives.

In Figure 1 we describe the simplified model for calculating the *interestingness* reputation score for *datum*. Our approach is informed by [1] which applies a similar methodology for surfacing interesting media objects. The score is an indicator of the likelihood that a particular datum has potential value. User interactions with the IMS are interpreted as *claim*s from Table 1. The figure shows *claim*s as they are consumed by various processes. The intermediate processes (boxes 7, 8, 9, and 10) compute normalized counts of the *claim* interactions. These counts are fed to the terminal process, *InterestingnessCustomMixer* (box 12) which scales and reduces the values into the scalar *interestingness* score. This score can be used as a predictor for search and recommendation systems. Omitted from this simplified model are lag and decay filters necessary to counteract volatility and freshness bias respectively [2].

5 Conclusion And Future Work

We have described a realistic blue print for a reputation system that is on the roadmap of our IMS. Once implemented we think that it will dramatically improve the quality of information that is retrievable by users, thus increasing its' effectiveness as a platform for information management and data science. We have left outcome analysis of the approach and results for a future paper. Also on the roadmap is a meaningful gamification system inspired by [4] to further enhance user engagement.

References

1. Butterfield, Daniel, et al. Interestingness Ranking of Media Objects. Yahoo! Inc., assignee. Patent US20060242139A1. 8 Feb. 2006. Print.
2. Farmer, F. Randall., and Bryce Glass. Building Web Reputation Systems. Sebastopol, CA: O'Reilly, 2010. Print.
3. Sveiby, K.E. Knowledge Management: Lessons from the Pioneers. Tech. Sveiby Knowledge Associates, 2001. Web. 12 Mar. 2013. http://www.sveiby.com/articles/KM-lessons.doc
4. Nicholson, Scott. Strategies for Meaningful Gamification: Concepts behind Transformative Play and Participatory Museums. Tech. Meaningful Play 2012 Web. 13 Mar. 2013. http://scottnicholson.com/pubs/meaningfulstrategies.pdf

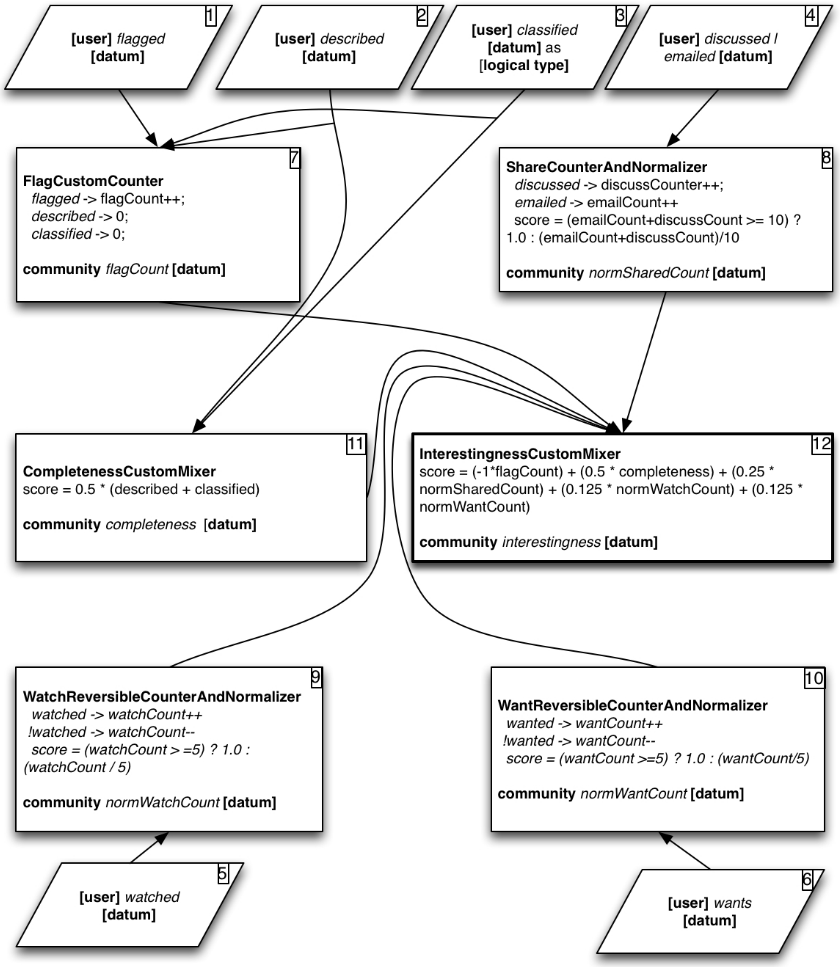

1
[user] *flagged* **[datum]**
2
[user] *described* **[datum]**
3
[user] *classified* **[datum]** as [**logical type]**
4
[user] *discussed* / *emailed* **[datum]**
7
FlagCustomCounter
flagged -> flagCount++;
described -> 0;
classified -> 0;
community *flagCount* **[datum]**
8
ShareCounterAndNormalizer
discussed -> discussCounter++;
emailed -> emailCount++
score = (emailCount+discussCount >= 10) ? 1.0 : (emailCount+discussCount)/10
community *normSharedCount* **[datum]**
11
CompletenessCustomMixer
score = 0.5 * (described + classified)
community *completeness* **[datum]**
12
InterestingnessCustomMixer
score = (-1*flagCount) + (0.5 * completeness) + (0.25 * normSharedCount) + (0.125 * normWatchCount) + (0.125 * normWantCount)
community *interestingness* **[datum]**
9
WatchReversibleCounterAndNormalizer
watched -> watchCount++
!watched -> watchCount--
score = (watchCount > =5) ? 1.0 : (watchCount / 5)
community *normWatchCount* **[datum]**
10
WantReversibleCounterAndNormalizer
wanted -> wantCount++
!wanted -> wantCount--
score = (wantCount >=5) ? 1.0 : (wantCount/5)
community *normWantCount* **[datum]**
5
[user] *watched* **[datum]**
6
[user] *wants* **[datum]**

Fig. 1. Interestingness Reputation Model for Datum

A Role for Provenance in Social Computation

Milan Markovic, Peter Edwards, and David Corsar

Computing Science & dot.rural Digital Economy Hub, University of Aberdeen,
Aberdeen, AB24 5UA
{m.markovic,p.edwards,dcorsar}@abdn.ac.uk

Abstract. We argue that existing systems to support social computation suffer from a lack of transparency and that this can be addressed by integrating provenance capture mechanisms into such systems. We discuss how Semantic Web technologies can be used to facilitate this, and how the provenance record could be used to support various forms of decision-making about tasks such as workforce selection.

1 Introduction

The widespread use of online interactive technologies has enabled new forms of computations based on the principles of collective intelligence [4, 2]. Robertson and Giunchiglia [4] define one such approach, social computation, as: *"a computation for which an executable specification exists, but the successful implementation of this specification depends upon computer-mediated social interaction between the human actors and its implementation"*. However, the use of humans in such computations introduces several issues including: reliability of workers, workforce selection, and quality of the generated results. To address these issues, current platforms such as Amazon's Mechanical Turk[1] provide basic reputation scores for workers based on acceptance of their product, tools for workforce selection based on worker's attributes (e.g. geolocation, qualifications) and means to assess results (e.g. by comparison with a gold standard).

We argue that recording the provenance of such activities and other aspects of social computation (e.g. formation of a group of participants) will increase the transparency of such systems, and so enable more sophisticated means of control. Such a provenance record would describe the activities performed throughout the computation, the entities (things) used and generated by those activities, and the agents associated with those activities [3]. This can then be used to enhance assessments of: workers reliability (e.g. forming beliefs about their trustworthiness based on their motives, past performance, capabilities, and relationships to trusted workers); results (e.g. by reconstructing and inspecting the events that lead to result generation); the process of the execution itself (e.g. how the group of workers necessary to complete the computation was formed).

The executable specification of a social computation can include *social properties* that define: "*the drivers for the adoption and spread of the computation*

[1] https://www.mturk.com/mturk/

through the social group with which it engages" [4]. For example, consider a system requiring a worker to provide a photograph of a current event, a social property could be: "to secure a reward, provide a photograph of the event or delegate the task to a trusted friend able to provide one". We argue that it is possible to use the provenance record generated by a social computation to infer worker compliance with those properties (i.e. to check if a worker's behaviour during the computation was consistent with such properties). Provenance information would also permit assessment of a worker's effect on the formation of the group of human participants performing the social computation (e.g. trusted worker Bob delegated the task to his friend Jack, whom he trusted and knew was at the event). Provenance can also be used to infer information about workers motive's (e.g. Bob was motivated to delegate the task in order to receive a reward). In addition, the provenance record can include information enabling the identification of worker's attributes such as their skills (e.g. Jack knew how to take a photograph) and capabilities (e.g. Jack was at the event and had a smartphone). We argue that using provenance to enable the kinds of reasoning highlighted here, would enhance the capabilities of decision-making processes such as trust assessment of workers and workforce selection.

2 Our Approach

We are investigating development of a provenance model for social computation that is aligned with Prov-DM[2], the current W3C provenance recommendation. An analysis of six platforms[3] identified aspects of social computation that a provenance model should describe: the task execution process; links to the social properties applicable for a task; how workers were motivated to participate; what skills and capabilities were associated with a worker when they performed a task; and constraints that were associated with the task description (e.g. requirement for photographs to be stamped by the device with its timestamp and geolocation). Prov-DM does not currently support explicit modelling of these aspects, and therefore one of our goals is to investigate and design a set of appropriate Prov-DM extensions to accommodate them.

Hendler and Berners-Lee [2] have previously argued that the fundamental role of Semantic Web technologies in social computation-like systems is to enable them to easily share data. For example, a process assessing the trustworthiness of Jack, based on the photograph he supplied, might consider Jack more trustworthy if it can determine that the picture was taken at the same time and place as the event. To do so, a system would compare the time and location associated with the photograph with those provided by a description of the event obtained from other data sources on the Web of Linked Data. In addition, such technologies provide a range of reasoning techniques that can be used to support

[2] http://www.w3.org/TR/prov-dm/

[3] These were: Amazon Mechanical Turk, CrowdFlower (http://crowdflower.com/), Zooniverse (http://zooniverse.org/), Passbrains (http://passbrains.com/); oDesk (http://odesk.com/); InnoCentive (http://innocentive.com/)

automated decision-making processes. For example, the fact that Bob delegated a task to Jack and did not provide the photograph, could result in in a naive system excluding Bob from future task assignments. However, Bob might be an important element contributing towards the formation of a group necessary to perform tasks (e.g. delegating trusted friends that provide results).

We argue that enhanced trust assessments of workers could lead to reductions in the number of workers required to perform additional result validation. Such validation steps are typical for current design patterns such as Find-Fix-Verify [1]. Furthermore, better understanding of the process of worker group formation and worker motivations could allow for the selection of smaller groups that perform computations resulting in the same or better results as larger groups. To evaluate our approach, we aim to develop a computational framework that utilises our extended provenance model, supported by semantic technologies. The framework should operate alongside existing platforms using an API to facilitate the capture and use of provenance.

3 Conclusions

In this paper we have argued that introduction of provenance capture mechanisms will not only increase transparency of social computations, but will also permit reasoning about aspects such as trustworthiness of workers and workforce recruitment. We suggest an approach to facilitate the capture and use of such provenance, with the support of semantic technologies and via extensions to Prov-DM. We are aware that there are a number of possible limitations of the proposed approach including scalability issues associated with processing of large provenance records; and difficulties in capturing certain aspects of provenance (e.g. worker's motivation). These remain interesting questions for our future work.

Acknowledgements The research described here is supported by the award made by the RCUK Digital Economy programme to the dot.rural Digital Economy Hub; award reference: EP/G066051/1

References

1. M.S. Bernstein, G. Little, R.C. Miller, B. Hartmann, M.S. Ackerman, D.R. Karger, D. Crowell, and K. Panovich. Soylent: a word processor with a crowd inside. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 313–322. ACM, 2010.
2. J. Hendler and T. Berners-Lee. From the semantic web to social machines: A research challenge for AI on the world wide web. *Artificial Intelligence*, 174(2):156–161, 2009.
3. L. Moreau and P. Missier. Prov-dm: The prov data model. W3C Recommendation (April 2012), http://www.w3.org/TR/prov-dm/, 2012.
4. D. Robertson and F. Giunchiglia. Programming the social computer. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1987), 2013.