Enhancing Named Entity Extraction by Effectively Incorporating the Crowd

Katrin Braunschweig, Maik Thiele, Julian Eberius, Wolfgang Lehner

Technische Universität Dresden
Faculty of Computer Science, Database Technology Group
01062 Dresden, Germany
{firstname.lastname}@tu-dresden.de

Abstract: Named entity extraction is an established research area in the field of information extraction. When tailored to a specific domain and with sufficient pre-labeled training data, state-of-the-art extraction algorithms have achieved near human performance. However, when presented with semi-structured data, informal text or unknown domains where training data is not available, extraction results can deteriorate significantly. Recent research has focused on crowdsourcing as an alternative to automatic named entity extraction or as a tool to generate the required training data. While humans easily adapt to semi-structured data and informal style, a crowd-based approach also introduces new issues due to monetary costs or spamming. We address these issues by combining automatic named entity extraction algorithms with crowdsourcing into a hybrid approach. We have conducted a wide range of experiments on real world data to identify a set of subtasks or operators, that can be performed either by the crowd or automatically. Results show that a meaningful combination of these operators into complex processing pipelines can significantly enhance the quality of named entity extraction in challenging scenarios, while at the same time reducing the monetary costs of crowdsourcing and the risk of misuse.

1 Introduction

Extracting named entities (NE) from unstructured data is an important NLP task in order to provide meaningful semantic search over the data. Such a search functionality is desirable not only for unstructured documents, but for a wide range of semi-structured documents, as well. However, many state-of-the-art NE extraction algorithms are specifically designed for unstructured text, relying on correct grammar and sentence structures. While these algorithms achieve near human performance on corpora that are similar to the training corpus regarding structure and content, they often show a much weaker performance on corpora with unfamiliar document and sentence structures. To improve the quality of NE extraction on these documents, better suited training data is required. However, obtaining a large enough training corpus labeled by experts is not only time

consuming but also expensive.
In contrast, recent research focused on NE crowdsourcing techniques as an alternative to automatic extraction or expert labeling. Therefore, crowdsourcing platforms, such as Amazon Mechanical Turk[1], or service providers, such as CrowdFlower[2], provide a set of tools to distribute tasks to a large number of people in exchange for monetary compensation. Offering a certain amount of flexibility regarding the individual task design, these platforms enable a wide spectrum of crowdsourcing tasks, ranging from simple tagging to more complex jobs such as text translation. However, not all tasks are equally suited for crowdsourcing. On the one hand, technical limitations of the platform itself can have a negative impact on the result quality. On the other hand, crowdsourcing platforms can be affected by spamming or misuse by crowd workers, leading to flawed results.
Previous results indicate that crowdcourcing NE extraction can achieve good results. However, while these works mostly focus on replacing automatic NE extraction by the crowd, irrespective of the costs, we focus on the combination of both approaches in order to enhance the overall extraction performance while keeping the costs minimal. Therefore, the main goal of this paper is to investigate different options for hybrid NE extraction with respect to their costs and effectiveness. With an experimental study on a corpus of semi-structured documents, we show how NE extraction can benefit most from the processing power of the crowd

The rest of this paper is organized as follows: In section 2, we analyze related work on both, automatic as well as crowd-based NE extraction, in order to identify the strengths and weaknesses of both alternatives. Based on these characteristics, we establish general options for hybrid NE extraction approaches in section 3. These options form the basis of our case study, which is presented in section 4. Finally, we summarize our findings and point out directions for future work.

2 Named Entity Extraction

Named entity extraction is an established part of many information extraction systems seeking to identify and label subsets of a text with categories such as *Person*, *Organization* or *Location*. In the following sections we distinguish between automatic approaches that have been studied extensively and the crowd-based extraction which emerged during the last three years.

2.1 Automatic Extraction

Many years of research have lead to great variety of automatic named entity extraction techniques. A common classification discriminates between rule-based

[1] www.mturk.com
[2] www.crowdflower.com

and learning-based approaches. Rule-based techniques, such as ANNIE [CMBT02] or SystemT [CKL+10], require the manual construction of grammar rules in order to extract NEs. In contrast, learning based approaches usually require a large annotated corpus in order to train a model based on discriminative features of entities in the corpus. In [NS07], Nadeau et al. present an extensive list of such learning-based approaches, including supervised approaches such as Conditional Random Fields or Support Vector Machines. While state-of-the-art algorithms tailored to a specific task achieve very good results, they are significantly less effective when presented with a new domain or text genre. Adaption to a new setting often requires new rules or new training data, which can be very expensive and time-consuming.

2.2 Crowd-based Extraction

As an alternative to automatic NE extraction, some researchers have focused their attention on crowdsourcing in order to obtain labeled training data from documents where classic extraction approaches fail and expert labeling is too expensive. Lawson et al. [LEPYY10] use Mechanical Turk to annotate emails, while Finin et al. [FMK+10] used both, Mechanical Turk and CrowdFlower to extract named entities from Twitter messages. Both text types differ in structure and style of writing from the classic type of newswire articles. In [VNFC10], Voyer et al. combine expert labeling with non-expert annotations from the crowd to obtain high-quality annotations at a lower expense. All of these examples use different task designs and layouts to present the task to the crowd. This highlights one of the main challenges when using a crowdsourcing approach: identifying the optimal task design. Layout, task granularity and additional parameters such as fee per unit, level of redundancy or batch size may influence the quality of the result, the overall costs as well as the processing time. The task of NE extraction can be seen as a single task or as a combination of two subtasks: 1) locating the span of a NE in the text, and 2) selecting the correct type of the NE. Finin et al. follow the single task approach in [FMK+10], using a form containing radio buttons to label each word in a Twitter message.[LEPYY10] and [YYSXH10] both use custom layouts which enable span selection to first locate potential NEs. Afterwards, they use drop-down menus and toggle buttons, respectively, to select the correct entity type. Voyer et al. [VNFC10] only leverage the crowd for the subtask of typing, leaving the task of span selection to experts in order to achieve a higher quality.
To improve the result quality, most crowdsourcing platforms offer some form of control mechanism. Mechanical Turk offers quality assessment tests, which workers have to pass, before being eligible to perform the actual tasks. In contrast, CrowdFlower uses gold data with pre-defined answers, which are mixed with the actual tasks to randomly check the quality of the worker's answers. If a worker fails too many gold questions, all of his answers are rejected. Another common control mechanism is inter-annotator agreement, where the same task is presented to several

(a) Extraction (b) Extension (c) Verification

Figure 1: Basic Operators

workers and an answer is only accepted if a sufficient number of workers agree. The level of redundancy as well as the level of agreement can be controlled by the task designer. Still, not all tasks are equally suited for these control mechanisms. Some tasks, such as span selection, are more likely affected by spam and misuse, since it is difficult to define correct gold data. Additionally, as noted in [LEPYY10], the fixed payment system of most crowdsourcing platforms can have a negative effect on NE extraction, as it does not encourage a high recall. Workers are paid per processed document, not for the number of NEs that have been extracted. For a worker, there is no difference between selecting one entity or ten. Mechanical Turk offers a bonus system, which allows the payment of a small bonus in addition to the regular fee in case a worker has performed very well and, for example, extracted a high number of NEs. However, such a bonus system is not available on all platforms.
Using crowdsourcing as an alternative to automatic processing introduces significent costs. Therefore, researchers have also studied different approaches to limit the costs using hybrid processing techniques. Wang et al. [WKFF12] used a hybrid approach for the task of entity resolution, with machines performing the initial matching and humans verifying only the most likely matches. Similarly, Demartini et al. [DDCM12] first use automatic matching for the task of entity linking and the crowd afterwards to improve the quality of the links.

3 Hybrid Extraction

Based on the different characteristics of automatic and crowd-based NE extraction, our goal is to further enhance the performance by combining both techniques into a hybrid approach. In addition to the quality, measured by precision and recall, we also focus on the effectiveness and quality-cost-ratio. It is important to note that the costs only refer to the monetary costs of the crowd-based tasks. The overall quality of a hybrid approach therefore depends on the three following parameters: the precision and recall as well as the total crowdsourcing costs. The optimization goal is to maximize precision and recall while keeping the costs minimal. Since it is very difficult to optimize all parameters simultaneously, we start with addressing each one individually.
In any case, the baseline for a hybrid extraction approach is formed by a classic NE extraction step, illustrated in Figure 1a. The input is an unlabeled document and

the output is a list of extracted entities or the labeled document. The extraction step itself can be either automatic or crowd-based. In case of a crowd-based approach, the costs can be calculated using the following cost function, where n_r is the redundancy level and n_b is the batch size.

$$C = \left\lceil \frac{U_{input} \times n_r}{n_b} \right\rceil \times c_a \tag{1}$$

The redundancy level determines how many workers are asked to perform a specific task, while the batch size determines how many of these tasks are grouped into a single assignment. Fees, depicted as c_a, are usually paid per assignment. U_{input} is the total number of individual units of work, which are presented to the crowd for processing. In most NE extraction applications, a unit of work is a document that needs to be labeled. If the extraction step is split into several consecutive crowdsourcing steps, the overall costs are determined by the sum of the individual costs.

3.1 Increasing Recall

Starting from an unlabeled document, an extraction step should always increase precision and recall. Obviously, the results of several extraction steps can be combined. However, it is hard to make any assumptions about the overall precision and recall. Instead, we focus on another type of processing step, which extends the list of named entities. Based on a list of entities extracted by a previous processing step, the extension step aims to extract additional entities that have not been identified before. As illustrated in Figure 1b, this type of processing step adds additional labels to a partially labeled document. Unless the respective extension algorithm is ineffective, we can expect overall recall to increase. Unfortunately, we can make no certain assumptions about its effecton precision. Again, the extension approach can be either automatic or crowd-based. The cost function remains the same as for the extraction step.

3.2 Increasing Precision

We also want to increase precision, which basically requires the removal of false positives from the set of extracted NEs. This can be achieved by adding a verification step, which is illustrated in Figure 1c. This step removes all labels from the document, which are identified as incorrect or conversely, which can not be identified as correct. Similar to the other processing steps, verification can be performed automatically or by the crowd. Given the applied technique is effective, this step increases the precision. Recall is either constant, if only false positives are removed, or decreases, if the approach is imprecise and removes not only false positives but also some true positives. Again, the cost of a crowd-based verification step can be calculated using

Equation 1, this time considering a single NE as a unit of work.

3.3 Reducing Costs

It is clear that the cost of a processing step based on crowdsourcing mainly depends on the number of units of work U_{input} that need to be processed. In some cases, U_{input} itself depends on the dataset (i.e. number of named entities contained in the dataset) as well as the performance of previous processing steps. While extraction and extension steps potentially increase the number of labeled NEs in the document, the opposite holds for verification steps. Verification reduces the number of labeled NEs and therefore also reduces the costs of subsequent crowdsourcing steps. Since every crowdsourcing step adds to the costs, it is important to only add them if they are effective and justify the investment.

We have identified three different generic classes of processing steps, each with different effects on precision, recall and costs. While extraction steps can be performed individually, the other two obviously only make sense in combination with anextraction step. We argue that combining these different processing steps into complex workflows can significantly improve the performance of NE extraction in challenging scenarios or domains, without necessarily increasing the total costs. To study the effect of the different classes in more detail and to confirm our theoretic assumptions, we conducted an extensive case study on real world data. We present the results of this study in the following sections.

4 Experimental Evaluation

4.1 Dataset

For our study we used a dataset containing 50 sets of metadata collected from the Open Data platform *data.gov.uk*. For each dataset a set of metadata is stored to support the retrieval of the dataset. An example is shown in Figure 2. Each set of metadata contains a list of properties, which we limited to the seven properties depicted in the example. The 50 documents have been selected from the platform with respect to their topic or publisher. We selected four topics with 10 related documents for each topic. A fifth group of 10 documents was added, containing randomly selected documents. To enable performance evaluation, all documents have been labeled manually. The gold data contains a total of 427 (162 distinct) named entities, including hierarchical NEs and abbreviations. Of these NEs 11 (11) are of type *Person*, 162 (47) of type *Location*, and 254 (104) of type *Organization*. Occurrence of a named entity is only counted once per property section. Organization is the most common type in the dataset. Person names are very rare. We found this dataset to be suitable for our study because it combines several features which cause

Figure 2: Example Dataset

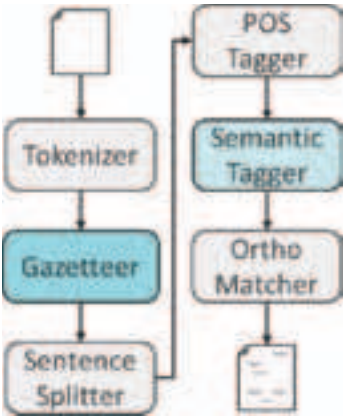


Figure 3: ANNIE Processing Pipeline

automatic extraction approaches to fail. First of all, it contains sections of different text genres. As shown in Figure 2, the section *Notes* contains classic unstructured text, while others, such as *Title* or *Description* contain only sentence fragments. The section *Tags* consist of a list of words or hyphenated phrases. Second, capitalization rules for proper names are not observed consistently. While some names appear in lowercase, other nouns are capitalized for emphasis.
Due to the described characteristics the named entity corpus is very ambitious. The dataset contains many difficult organization names, including hierarchical NEs. For example, the organization name "Taunton and Somerset NHS Foundation Trust" also contains the location names "Taunton" and "Somerset" as well as the organization name "NHS". Additionally, the corpus contains a large number of abbreviations of both location and organization names, such as "DEFRA". In some cases, nonstandard shortened versions of organization names are used, which are very difficult to detect. Due to the high percentage of hierarchical NEs and abbreviations in the corpus, it is rather unlikely to achieve a very high recall.

4.2 Crowdsourcing Environment and Setup

All crowdsourcing tasks have been created using the CrowdFlower Builder and executed on Amazon Mechanical Turk via CrowdFlower. We tried to keep the external conditions of our experiments as stable as possible, posting new tasks at the same time of day (2-3 pm, GMT) to the same platform (Amazon Mechanical Turk). The list of countries, which specified which workers were eligible to perform our tasks, was also kept unchanged throughout our study. We focused on a list of predominantly English speaking countries to reduce the risk of misinterpreting the task description.

We also kept the internal setting constant for all experiments. If not stated otherwise, we set the redundancy level n_r to 3 and batch size n_b to 5. The minimum inter-annotator agreement was set to 2, meaning that at least 2 out of the 3 workers performing the same task would have to agree on the answer, otherwise the result was discarded as *undecided*. Due to the costs involved, all crowdsourcing tasks were performed only once per experiment. Therefore, the results do not represent the statistic mean.

4.3 Extraction Algorithms

As the baseline of our experiments, we tested both automatic as well as crowd-based extraction individually.

4.3.1 Automatic Extraction

As an example of an automatic NE extraction algorithm, we used the rule-based approach ANNIE [CMBT02]. We decided on a rule-based technique, because the interpretation of the algorithm's behavior is easier for rules than for statistic approaches. It is important to note that statistic approaches, which were not specifically trained for our corpus, did not achieve significantly better results than the rule-based approach. ANNIE extracts entities from documents using a pipeline of different processing steps, as illustrated in Figure 3. Both, gazetteer lists and regular expression-like rules, are used to identify NEs of various types. The performance results for ANNIE are presented in Table 4.

4.3.2 Crowd-based Extraction

As mentioned in Section 4.3.2, there is a wide range of options, how to present the overall task of named entity extraction to the crowd. The task design has a direct impact on the monetary costs as well as potential impact on the result quality. We studied different layout designs and settled on splitting the task into two subtasks: 1) the location of NEs, and 2) the type selection.
For the first subtask, we asked workers to mark the span of potential NEs in each document, without the need to decide on a type. Due to the size of each document, we did not batch several task into a single assignment for this step. The layout of this task is shown in Figure 6a. In the second task, we asked workers to select the correct type for each NE from a drop-down menu. To provide some context, we display the section of the document, which contained the NE. The layout for this tasks is shown in Figure 6b. Due to its smaller size, we batched the second task, with each assignment containing 5 NEs. For each task we used a redundancy level of 3. The type selection required an inter-annotator agreement of at least 2. However, for the complex task of marking the span of an NE in the text, we did not take agreement between annotators into account and accepted all labels to increase

	ANNIE		Crowd	
Topic	P	R	P	R
CO	**0.67**	**0.59**	0.56	0.17
DEFRA	0.57	0.27	**0.76**	**0.48**
MISC	**0.57**	**0.55**	0.52	0.34
NERC	0.72	0.33	**0.77**	**0.42**
NHS	0.65	**0.42**	**0.7**	0.29
All	0.64	**0.42**	**0.68**	0.34

Figure 4: Precision (P) and Recall (R) for NE extraction using ANNIE and the crowd, respectively.

Experiment	P	R	C
Abbreviations	0.67	0.5	0
Crowd (Tags)	0.65	0.48	70
Crowd (All)	0.63	0.61	285

Figure 5: Precision (P), Recall (R) and Cost (C) of Extension Steps in Combination with ANNIE.

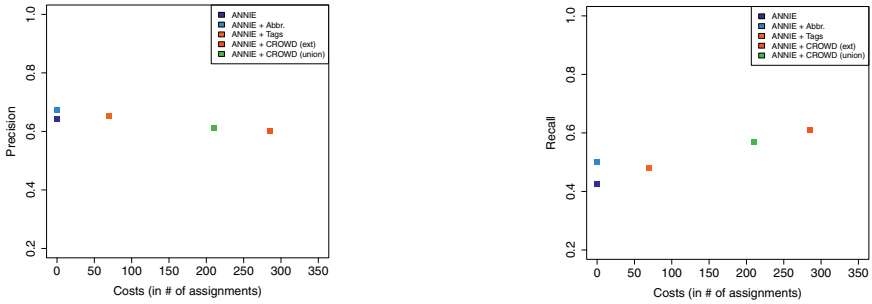
recall.
The span selection task proved very difficult with respect to quality control and spamming. Due to access constraints on Mechanical Turk, we could not make use of the bonus system to increase recall. It was also difficult to incorporate the control mechanisms provided by CrowdFlower. Therefore, we had to perform the labeling step without any quality control mechanisms, which resulted in a high number of empty documents in the result set. For the classification step, we added a set of gold data to enable quality control. The performance of this experiment is also presented in Table 4. The crowd-based extraction requires the processing of 210 assignments by the workers on Mechanical Turk. In our study, we paid workers \$0.01 or \$0.02 per assignment, depending on the difficulty of the task.
Overall, the crowd-based approach achieved a slightly higher precision than ANNIE, but at a lower recall. Still, the results are very similar. However, analyzing the performance of each approach separately for each of the 5 groups of documents described in Section 4.1 reveals significant performance differences. Especially the performances for groups *CO* and *DEFRA* indicate that the different extraction approaches do not behave similarly for all document. Based on this observation, we expect a hybrid approache to perform better than each individual approach. Even a naive union of both result sets achieves a precision of 0.61 and a recall of 0.57.

4.4 Extension Approaches

To increase the recall of our NE extraction task, we studied three extension approaches, an algorithmic approach and two crowd-based approaches. For the algorithmic approach, we used the high number of abbreviations in the corpus as motivation. Both extraction approaches described in the previous section, identified only a small percentage of these abbreviations. Therefore, we used a simple algorithm to identify additional abbreviations. The algorithm takes previously labeled entities of types *Location* and *Organization*, which contain at least 2 words, as input

(a) Span Selection (b) Type Selection (c) Type Verification

Figure 6: CrowdFlower Tasks



(a) Cost and Precision (b) Cost and Recall

Figure 7: Extension Steps

and creates several potential abbreviations following different patterns. For the name "Department for Environment, Food and Rural Affairs", for example, the following five patterns were created: "DEFRA", "DFEFARA", "defra", "dfefara", and "Defra". Afterwards the algorithm searches for occurences of these abbreviations within the document where the original NE was found and assigns the correct type to each occurence. The performance of this extension step (in combination with ANNIE as the primary extraction algorithm) is presented in Table 5.

The first crowd-based extension approach is similar to the crowd-based extraction approach described in Section 4.3.2. However, it is not performed for all documents, only for sections of documents, where a previous extraction step failed to find any NEs. We noticed that ANNIE could not deal with the format of the *Tags* section in the documents, so we asked the crowd to label only these sections. Again, we used the same processing pipeline consisting of two separate crowd tasks as before. Due to the smaller size of the sections, we used a batch size of 5 for both steps. Performance and costs of this approach are also presented in Table 5.

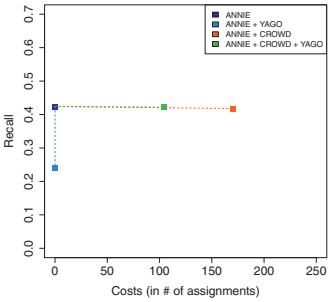
The second crowd-based approach is also very similar to the crowd-based extraction, but with minor differences. This time, previously labeled NEs are marked in the text and the workers are asked to only mark NEs, which have not been labeled

before. The parameters n_r and n_b are the same as in Section 4.3.2, but it is obvious, that this is a more difficult task and likely to produce less precise results. The results in Table 5 show a slight decrease in precision. However, the recall increased significantly compared to the baseline.
Figures 7a and 7b compare the effects of each of the proposed extension techniques on precision and recall, respectively. They also show the costs involved to perform each step. The results confirm the assumption that while each extension step increases recall to some extend, we cannot make any definite assumptions about the effect on precision.

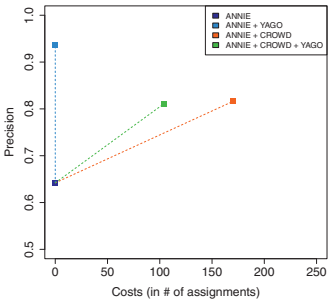
4.5 Verification

As described in Section 3.2, in order to increase precision, we incorporate a verification step which needs to answer two questions: 1) Is the NE in question a proper name, and 2) has the NE been typed correctly? Only if the answers to both questions are yes, the NE is added to the result set. Again, we have tested both, an automatic and a crowd-based approach.
The automatic verification technique is inspired by related approaches in the field of question answering, where evidence from external sources, such as knowledge bases, is collected in order to verify the answer type. In a similar fashion, we collect evidence from the YAGO knowledge base [SKW07] to verify the types of our named entities. YAGO is a structured knowledge base, which consolidates information from various sources, including Wikipedia, WordNet and GeoNames and contains a total of $9,756,178$ entities[3]. The knowledge base can be queried using SPARQL queries. For each type (i.e. Person, Location, Organization), we defined several different queries. We accepted a named entity as correct if at least one query returned a positive match.
For the type *Person*, we searched YAGO for entities of type *person*, whose preferred name matched the NE. Since YAGO contains multiple types with the name *person*, we limited our search to the type extracted from WordNet. Additionally, we tried to extract the first name as well as the last name from the NE and searched YAGO for entities with attributes *hasGivenName* and *hasLastName*.
For type *Location*, we investigated the YAGO WordNet part for entities whose type was *location* or a subclass of *location*, and whose preferred name matched the NE. We also queried YAGO for entities with a *yagoGeoEntity* type or types of the subclasses of *yagoGeoEntity*.
For the type *Organization*, we first looked for entities, whose type was a subclass of YAGO type *organization*. However, for NEs of type *Organization*, this YAGO based technique turned out to be too imprecise, accepting too many false positives. Therefore, we used this approach only for NEs of type *Person* or *Location* in our experiment. Although this approach is expected to significantly increase precision, it also has the disadvantage that named entities from the corpus, which do not appear

[3] www.mpi-inf.mpg.de/yago-naga/yago/statistics.html



(a) Costs and Recall



(b) Costs and Precision

Figure 8: Verification Steps

in any knowledge base, are dismissed, reducing recall significantly. That means that finding no supporting evidence in the knowledge base is no clear indication that the NE is a false positive. The results in Table 10 clearly show the negative impact on recall.

For the crowd-based verification, we presented workers with the NE, its type and its context. We then asked them to verify if the entity in question was of the specified type. The task layout is illustrated in Figure 6c. The task was performed per NE extracted from the documents, using the same parameters as described in Section 4.2. Again, we added gold data for quality control. The results of this verification step (in combination with ANNIE) are shown in Table 10. As expected, the verification step significantly increases the precision compared to the baseline experiment, although to a lesser extend than the YAGO based approach. But, it does not have a negative impact on recall.

As noted in Section 3.3, the costs of a crowdsourcing step depend on the number of units that need to be processed, which in this case is the number of NEs that need to be verified. Our results show that the YAGO based approach can verify some of the NEs with very high precision, but misses some of the correct NEs in the process. We can use this characteristic to our advantage, by combining both verification techniques. As shown in Figure 9, we can use YAGO first to verify all NEs and then use the crowd to check those NEs again, which could not be verified in the first step. Figures 8a and 8b show that this combined approach achieves the same precision and recall as the crowd based verification, but at a significantly lower cost.

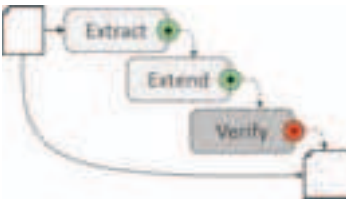
4.6 Complex Workflows

After studying the performance of the three classes of processing steps individually, we now take a look at what performance can be achieved when combining multiple

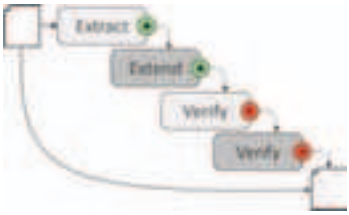
Figure 9: Combination of Automatic (white) and Crowd-based (gray) Verification

Experiment	P	R	C
YAGO	0.94	0.24	0
Crowd	0.82	0.42	170
YAGO + Crowd	0.82	0.42	104

Figure 10: Precision (P), Recall (R) and Cost (C) of Verification Steps in Combination with ANNIE.



(a) Workflow 1



(b) Workflow 2

Figure 11: Complex Workflows with Automatic (white) and Crowd-based (gray) Processing Steps

processing steps into complex workflows. For the first experiment, we combined the ANNIE extraction step with the automatic extension that searches for abbreviations in the document, as well as a crowd-based verification. The pipeline is shown in Figure 11a. Precision, recall and costs for this experiment are presented in Table 12. For the second experiment, we combined ANNIE with a crowd-based extension step as described in Section 4.4. We then used both YAGO and the crowd to verify the results. This workflow contains two crowd-based processing steps, which naturally leads to higher costs. The processing pipeline is shown in Figure 11b and precision, recall and costs for this experiment are also included in Table 12.

The results show that combining several processing steps into complex workflows can significantly improve the overall performance for NE extraction tasks. Figure 13 illustrates the different effects each processing step has on precision and recall. We can see in both experiments, that the extension step increases the recall, while the verification step increases precision. While the first workflow achieves a 30% increase in precision and a 15% increase in recall, compared to the baseline (i.e. ANNIE), the second workflow achieves a 44% increase in recall but only 14% increase in precision. So in a scenario where quality is more important than quantity, the first workflow would be more suitable. If, however, quantity trumps quality, the second workflow would be the better choice. Of course, the results of these experiments do not present the upper limit that can be achieved by combining multiple processing steps.

Compared to the use of crowdsourcing for NE extraction as a simple alternative to

	Workflow 1	Workflow 2
Precision	**0.84**	0.73
Recall	0.49	**0.61**
Costs	191	480

Figure 12: Precision, Recall and Cost of Complex Workflows.

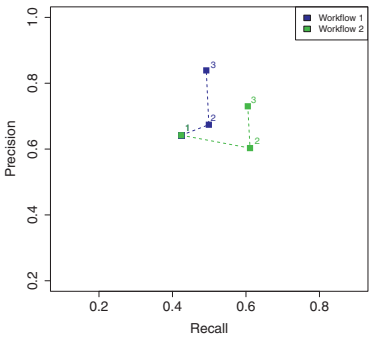


Figure 13: Change in Precision and Recall after Extraction (1), after Extension (2), and after Verification (3)

automatic extraction techniques, we can achieve a much better result for our corpus, using a complex pipeline of processing steps. Although the moderate performance of our crowd-based extraction is partially due to the absence of both, quality control mechanisms and a bonus system, we can still achieve a much better result at lower costs than the original crowd-based extraction. Using workflow 1, we can achieve higher precision and recall at a cost of 191 assignments, compared to the 210 assignments required for the crowd-based extraction.

5 Conclusion and Future Work

Adapting the task of named entity extraction to new genres and domains can be a great challenge, as it often requires sufficient labeled training data for testing and evaluation. Crowdsourcing has been proposed as a cheaper alternative to expert labeling. However, in some cases, the performance of a crowd-based extraction does not meet the expectations, as shown in our example. To address this issue, we identified several additional techniques to enhance the extraction performance. By combining automatic and crowd-based techniques into a complex processing pipeline, we could achieve significantly better results for our test corpus than automatic or crowd-based extraction achieved individually. By investing in crowd-based extension and verification instead, we managed to achieve an increase in both, precision and recall. In some cases, we also reduced the overall costs.
The combination of different processing steps allows for a great variety of different processing pipelines. While some showed a higher impact on precision, others effected recall more. This shows that the extraction pipelines can be tailored to the specific requirements of the application.
Based on our proposed set of operators we would like to extent the approach with a

comprehensive cost and confidence model which allows us to estimate the expected quality-cost-ratios of different workflow alternatives. Given such a model we could choose the best workflow for given requirements on precision, recall and costs, in many ways similar to the hybrid query execution plans proposed by Franklin et al. in [FKK+11].

References

[CKL+10] L. Chiticariu, R. Krishnamurthy, Y. Li, S. Raghavan, F. R. Reiss, and S. Vaithyanathan. SystemT: an algebraic approach to declarative information extraction. In *ACL '10*, pages 128–137, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[CMBT02] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *ACL '02*, 2002.

[DDCM12] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux. ZenCrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 469–478, New York, NY, USA, 2012. ACM.

[FKK+11] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. CrowdDB: answering queries with crowdsourcing. In *SIGMOD '11*, pages 61–72, New York, NY, USA, 2011. ACM.

[FMK+10] T. Finin, W. Murnane, A. Karandikar, N. Keller, J. Martineau, and M. Dredze. Annotating named entities in Twitter data with crowdsourcing. In *CSLDAMT '10*, pages 80–88, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[LEPYY10] N. Lawson, K. Eustice, M. Perkowitz, and M. Yetisgen-Yildiz. Annotating large email datasets for named entity recognition with Mechanical Turk. In *CSLDAMT '10*, pages 71–79, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[NS07] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January 2007. Publisher: John Benjamins Publishing Company.

[SKW07] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A Core of Semantic Knowledge. In *WWW '07*, New York, NY, USA, 2007. ACM Press.

[VNFC10] R. Voyer, V. Nygaard, W. Fitzgerald, and H. Copperman. A hybrid model for annotating named entity training corpora. In *LAW IV '10*, pages 243–246, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[WKFF12] J. Wang, T. Kraska, M. J. Franklin, and J. Feng. CrowdER: Crowdsourcing Entity Resolution. In *VLDB*, 2012.

[YYSXH10] M. Yetisgen-Yildiz, I. Solti, F. Xia, and S. R. Halgrim. Preliminary experience with Amazon's Mechanical Turk for annotating medical named entities. In *CSLDAMT '10*, pages 180–183, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.