Accuracy Metrics for Entity Extraction

Kelly Enochson, PhD; Gregory Roberts
Rosoka Software, Inc. 950 Herndon Parkway, Suite 370, Herndon, VA 20170

Abstract

Entity extraction software is typically evaluated based on the widely-accepted accuracy metrics of precision, recall, and F-measure. These metrics are certainly useful but limited in their scope. Additional factors including the types of errors, the cost of different error types, the facility of making changes to the system, and the efficiency of the system compared to human tagging should also be incorporated when evaluating entity extraction software. This paper illustrates the need for these additional factors and demonstrates how they can be implemented in evaluation.

1 Introduction

Natural language processing systems, including entity extraction tools, are typically evaluated using the metrics of precision, recall, and F-measure. *Precision* measures the proportion of extracted entities that are, in fact, entities; *recall* measures the proportion of actual entities in the data that are successfully extracted by the software; *F-measure* represents a weighted mean of precision and recall.

While it is certainly useful to identify accuracy metrics that can be uniformly applied across systems, as Powers (2011) notes, there is bias inherent in these metrics. For example, these metrics exclude the classification of negatives (ie., *true negative rate*, which represents the proportion of real negative instances that are correctly identified as negative, and *true negative accuracy*, which represents the proportion of instances identified by the software as negative that are, in fact, negative), which can have a disproportionate cost given a skewed distribution. For example, if there are many more negative instances in the data than positive, a high recall score will be misleading without the additional inclusion of the true negative rate. Additionally, as Chincor et al. (1993) notes, F-measure assumes a uniform cost across error types, and for many entity extraction users this is not the case. For example, it may be far worse for a user to miss the extraction of a person name than to extract a string that isn't a person name. For the same user, this may only apply to person names, with precision being favored for other entity types. F-measure can be weighted to prioritize recall over precision, but the beta value is uniform across entity types and subtypes, attributes, data types, etc. and may not accurately reflect the needs of the user.

It is also useful when assessing entity extraction software performance to consider the error itself. An extracted entity that is clearly incorrect may be less problematic than an extracted entity that is wrong but not obviously so; the latter is more likely to slip through the cracks of quality control. Moreover, the ease with which users can modify the extraction engine in response to identified errors is an important factor to consider as well. Errors that occur in a tool that allows users to quickly update extraction rules are less problematic than errors that arise from closed tools or tools that require significant technical expertise to modify.

Finally, it is important to consider the accuracy of human evaluators when assessing the accuracy of NLP software. In the MUC-3 evaluation, Chincor et al. (1993) found that well-trained human evaluators agreed in their document tagging at a rate of about 90%, and required several days to complete the manual tagging of 100 documents (p. 418). The values of precision and recall assigned to a software system will show variability correspondent to this human evaluator variability.

A comprehensive assessment of the effectiveness and efficiency of entity extraction software should evaluate accuracy metrics, error type and cost, and level of effort to fix problems. This paper demonstrates how these measures can be applied to an assessment of PERSON entity extraction.

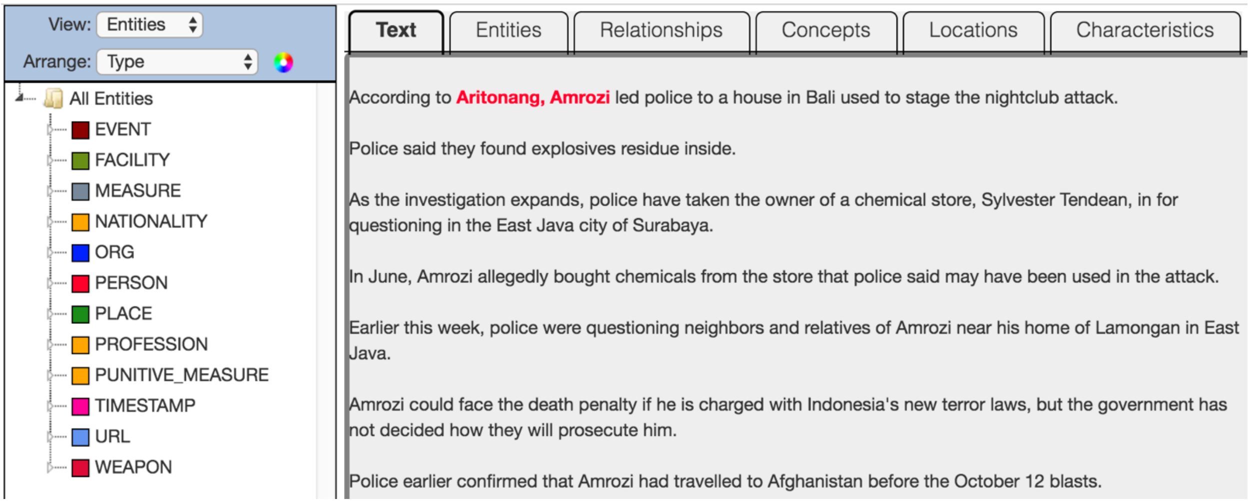
2 Rosoka Software PERSON Extraction

Entity extraction systems should be evaluated based on how accurate the output is, how easily the errors can be identified post-extraction, and the level of effort required to improve the system based on the errors. The current demonstration uses a corpus of customer data to assess these factors for Rosoka Software's extraction of person names.

Rosoka Extraction is a multilingual entity and relationship extraction tool. It extracts the names of people—among other entities—from unstructured text documents, and provides both surname and pronominal anaphora resolution. For example, if the name in a text is *Kelly Enochson*, the tool will extract this as PERSON with the attributes given_name="kelly" and sur_name="enochson", and will recognize that *Dr. Enochson*, *Enochson*, and *she* are all references to that same entity. Names that tend to be in different formats, such as Asian names in the format *Surname Given name*, and bibliographic references in the format *Surname, Given name*, will also be extracted via complementary rules in Rosoka's extraction engine.

The data used for this demonstration come from a corpus of customer data consisting of 61,394 news documents in English. The documents were processed using Rosoka Extraction Series 5. From these documents, 873,852 total PERSON entities were extracted, of which 205,726 were unique PERSON entities. Each instance of an extracted PERSON entity was vetted by a computational linguist for accuracy. Of the 873,852 extracted instances, Rosoka Extraction was correct 855,156 times, corresponding to a precision score of 97.86 and an error of 2.139, well above the inter-rater reliability reported in Chincor et al. (1993). Because the time required to manually tag this quantity of data is prohibitive, recall metrics are not available.

The most common extraction errors among these data fell into two categories: identifying two different names separated by a comma as a PERSON in the format *surname, given name*, as shown in Figure 1, and extracting only one part of a larger person name, as shown in Figure 2. In Figure 1 the name *Amrozi* is a single name alias that the software erroneously identified as a given name in this context. In Figure 2 the name *Matori Abdul Djalali* is extracted correctly, but the subsequent anaphoric reference *Dalali* is extracted as a separate person.



According to **Aritonang, Amrozi** led police to a house in Bali used to stage the nightclub attack.

Police said they found explosives residue inside.

As the investigation expands, police have taken the owner of a chemical store, Sylvester Tendean, in for questioning in the East Java city of Surabaya.

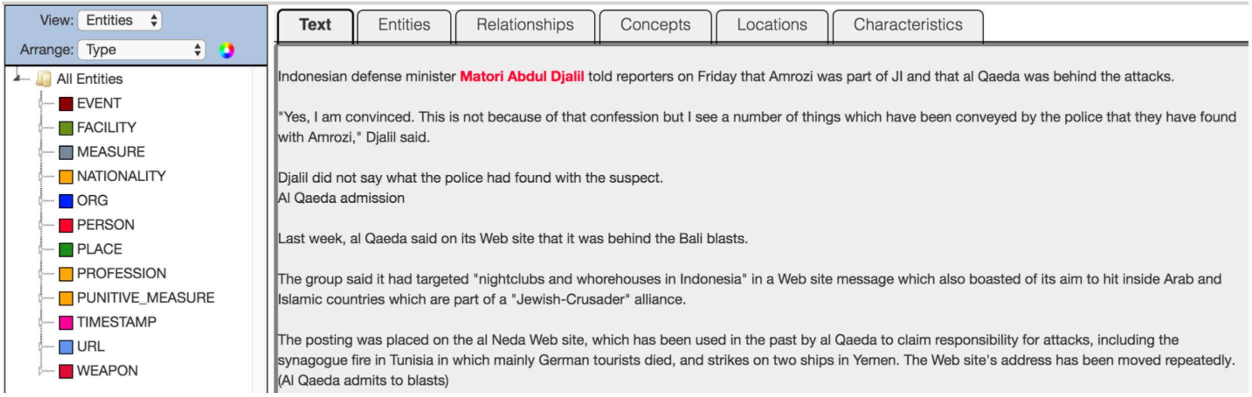
In June, Amrozi allegedly bought chemicals from the store that police said may have been used in the attack.

Earlier this week, police were questioning neighbors and relatives of Amrozi near his home of Lamongan in East Java.

Amrozi could face the death penalty if he is charged with Indonesia's new terror laws, but the government has not decided how they will prosecute him.

Police earlier confirmed that Amrozi had travelled to Afghanistan before the October 12 blasts.

Figure 1: Two names separated by a comma.



Indonesian defense minister **Matori Abdul Djalil** told reporters on Friday that Amrozi was part of JI and that al Qaeda was behind the attacks.

"Yes, I am convinced. This is not because of that confession but I see a number of things which have been conveyed by the police that they have found with Amrozi," Djalil said.

Djalil did not say what the police had found with the suspect.
Al Qaeda admission

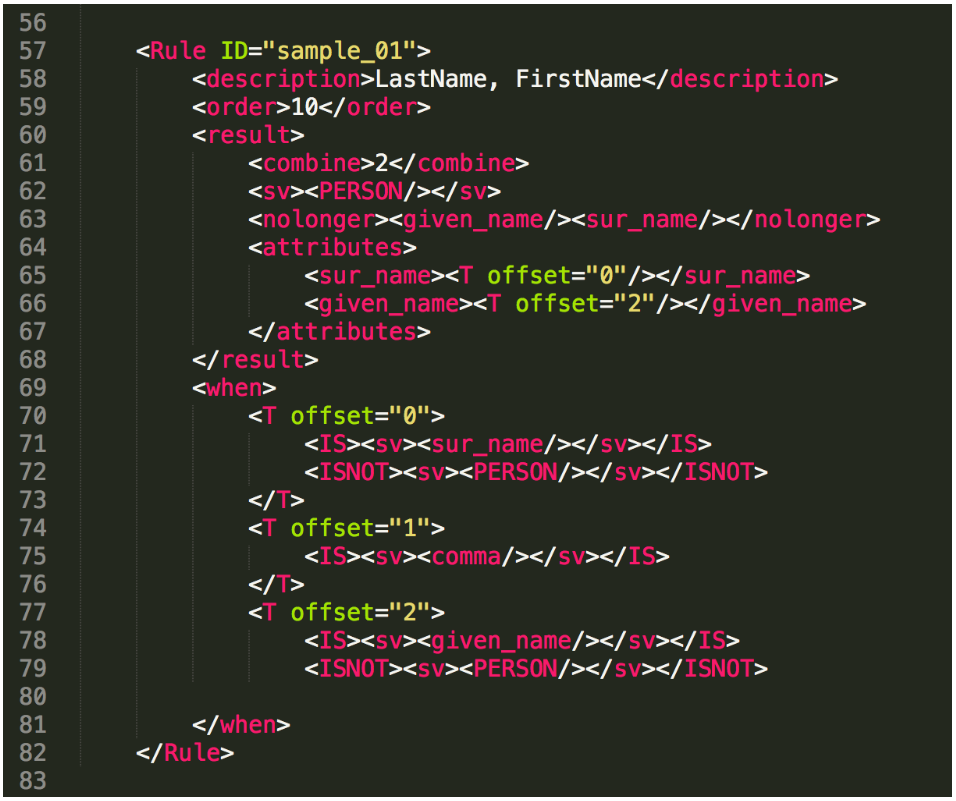
Last week, al Qaeda said on its Web site that it was behind the Bali blasts.

The group said it had targeted "nightclubs and whorehouses in Indonesia" in a Web site message which also boasted of its aim to hit inside Arab and Islamic countries which are part of a "Jewish-Crusader" alliance.

The posting was placed on the al Neda Web site, which has been used in the past by al Qaeda to claim responsibility for attacks, including the synagogue fire in Tunisia in which mainly German tourists died, and strikes on two ships in Yemen. The Web site's address has been moved repeatedly. (Al Qaeda admits to blasts)

Figure 2: Partial name extraction.

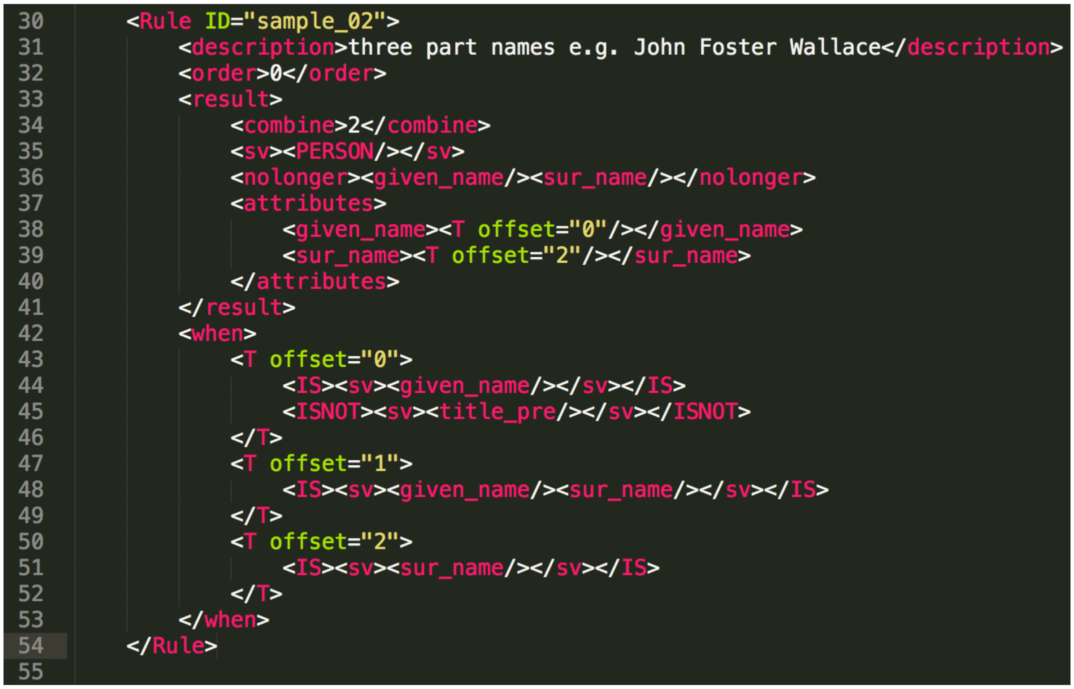
These two error types account for 14,498 of the 18,696 errors, or 77.5% of the total errors. Rosoka Software allows users to modify the knowledge base (including the ontology, rules, lexicons, etc.) to effect changes to extraction results. Making the necessary changes to remedy the two types of errors described here involves modifying two rules, one to include a Boolean NOT parameter, and one to reassign given name and surname attributes so names like that in (2) are extracted via surname anaphora resolution. Figure 3 shows the NOT parameter in lines 72 and 79. Figure 4 shows attribute assignment in lines 38 and 39. Identifying the problematic rules, making changes, and testing the results takes about 15 minutes, and amounts to a data change, not a code change. The system does not need to be recompiled, although the documents do need to be reprocessed. Once these changes are implemented, the precision score increases to 99.5.



```
    <Rule ID="sample_01">
        <description>LastName, FirstName</description>
        <order>10</order>
        <result>
            <combine>2</combine>
            <sv><PERSON/></sv>
            <nolonger><given_name/><sur_name/></nolonger>
            <attributes>
                <sur_name><T offset="0"/></sur_name>
                <given_name><T offset="2"/></given_name>
            </attributes>
        </result>
        <when>
            <T offset="0">
                <IS><sv><sur_name/></sv></IS>
                <ISNOT><sv><PERSON/></sv></ISNOT>
            </T>
            <T offset="1">
                <IS><sv><comma/></sv></IS>
            </T>
            <T offset="2">
                <IS><sv><given_name/></sv></IS>
                <ISNOT><sv><PERSON/></sv></ISNOT>

        </when>
    </Rule>
```

Figure 3: Adding a NOT parameter.



```
    <Rule ID="sample_02">
        <description>three part names e.g. John Foster Wallace</description>
        <order>0</order>
        <result>
            <combine>2</combine>
            <sv><PERSON/></sv>
            <nolonger><given_name/><sur_name/></nolonger>
            <attributes>
                <given_name><T offset="0"/></given_name>
                <sur_name><T offset="2"/></sur_name>
            </attributes>
        </result>
        <when>
            <T offset="0">
                <IS><sv><given_name/></sv></IS>
                <ISNOT><sv><title_pre/></sv></ISNOT>
            </T>
            <T offset="1">
                <IS><sv><given_name/><sur_name/></sv></IS>
            </T>
            <T offset="2">
                <IS><sv><sur_name/></sv></IS>
            </T>
        </when>
    </Rule>
```

Figure 4: Assigning attributes.

The rules are written to extract as efficiently as possible across a wide range of data types, topics, and languages. These changes serve to tune the out-of-the-box rules to the specific data in this use case.

In addition to the minimal effort required to effect changes improving extraction results, it is also important to note that the extraction errors in this data set tend to be obvious errors that are likely to be identified by an analyst or post-extraction quality control procedures. For example, incorrectly identified PERSON entities like “Tuesday Trump” are clearly incorrect, allowing them to be easily caught and fixed in the system.

3 Conclusion

Entity extraction software is typically evaluated based on the metrics of precision, recall, and F-measure. This paper argues that, while these metrics can be useful, there are additional meaningful factors that should be evaluated as well. Software that can be easily modified and customized allows users to fix errors as they are identified. Software that produces errors that are easily identified allows users to evaluate results efficiently and implement necessary changes more effectively. The speed with which entity extraction software can process documents likely makes the software far more efficient than using analysts or linguists to manually tag documents. Human accuracy is demonstrated to be around 90% (Chincor et al, 1993), so software that performs at an equal or higher accuracy level than this is more effective and efficient than human performance. When comparing two disparate extraction systems, measuring precision, recall, and F-measure is likely to be meaningless. A more meaningful approach is to measure the level of effort required to modify the out-of-the-box product to suit a specific use case. For example, how much time, what level of skill and training, and how many support hours are necessary to add a new entity type? As this paper demonstrates, a tool that can be quickly and easily modified based on customer data will require relatively little effort to effect demonstrable changes in accuracy and effectiveness.

References

Chinchor, N. (1991, May). MUC-3 linguistic phenomena test experiment. In *Proceedings of the 3rd conference on Message understanding* (pp. 31-45). Association for Computational Linguistics.

Chinchor, N. A. (1998, October). MUC/MET evaluation trends. In *Proceedings of a workshop on held at Baltimore, Maryland: October 13-15, 1998* (pp. 235-239). Association for Computational Linguistics.

Chinchor, N., Lewis, D. D., & Hirschman, L. (1993). Evaluating message understanding systems: an analysis of the third message understanding conference (MUC-3). *Computational Linguistics*, *19*(3), 409-449.

Daiber, J., Jakob, M., Hokamp, C., & Mendes, P. N. (2013, September). Improving efficiency and accuracy in multilingual entity extraction. In*Proceedings of the 9th International Conference on Semantic Systems* (pp. 121-124). ACM.

Powers, D.M.W. (2011). Evaluation: from Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation. *Journal of Machine Learning Technologies*, 2(1), 37-63.

Rosoka Software, Inc. (2010). Rosoka Extraction (Version 5) [computer software]. Retrieved from www.rosoka.com