Complementarity, F-score, and NLP Evaluation

Leon Derczynski
University of Sheffield
S1 4DP, UK
leon.derczynski@sheffield.ac.uk

Abstract

This paper addresses the problem of quantifying the differences between entity extraction systems, where in general only a small proportion a document should be selected. Comparing overall accuracy is not very useful in these cases, as small differences in accuracy may correspond to huge differences in selections over the target minority class. Conventionally, one may use per-token complementarity to describe these differences, but it is not very useful when the set is heavily skewed. In such situations, which are common in information retrieval and entity recognition, metrics like precision and recall are typically used to describe performance. However, precision and recall fail to describe the differences between sets of objects selected by different decision strategies, instead just describing the proportional amount of correct and incorrect objects selected. This paper presents a method for measuring complementarity for precision, recall and F-score, quantifying the difference between entity extraction approaches.

Keywords: Evaluation, F-Score, Ensembles

1. Introduction

Many NLP systems are evaluated using F-score, which describes system performance using a scale from zero to one. However, F-score lacks detail. When two approaches to an NLP problem achieve similar F-scores, they are not necessarily successful at the same kind of thing. This paper attempts to address this lack of detail, describing some of its impacts and putting forward an evaluation measure to be used alongside F-score that alleviates some of these problems.

F-score itself is derived from two summary measures: precision and recall. Precision describes the proportion of entities – e.g. mentions of people, events, or any given target phenomenon – which a system returns that are correct. Recall describes the proportion of all entities that potentially should be found, that a given system actually returns. Like F-score, these two are also summary measures, and so suffer from a similar lack of detail. When one NLP system achieves higher recall or precision than another, it does not imply that the better-scoring system accurately reproduces the results of the other and then exceeds them; rather, just the overall selection of entities is more precise, or more comprehensive, in some way. And indeed, high precision or recall can be achieved superficially through extreme – and often useless – conditions, like "return everything".

None of these measures can tell us much about the strategies that systems adopt. Importantly, they also cannot tell us about the differences in mistakes that systems make. Identifying strongly-different systems tells us which comparisons are interesting to make, and which data we should look at in order to learn the most about a system, task or dataset. Seeing the impact that different strategies have can be interesting from a qualitative point of view: one system may do a lot better on data that is rich in a particular kind of linguistic phenomenon, for example. Also, it is interesting to see the impact that different kinds of prior knowledge and models have, through for example changes in feature representation – two systems may reach the same F-score, but on very different examples, depending on what information they use.

If systems make the same mistakes, there is little uniqueness that sets the systems apart, and little to be gained by combining them. Conversely, if they make reasonably different mistakes, one may be able to profit from e.g. combining them and reducing the number of cases where they are both wrong, and conducting analyses. This difference, hidden by F-score, can be measured in terms of **complementarity**.

2. Complementarity

Complementarity (Brill and Wu, 1998) is a measure of the difference in decisions made by two systems that assign labels. It represents the amount of times where one system is wrong that the other is correct. This is referred to as the complementary rate, and works as follows. Given systems A and B, $Comp(A, B)$ is the proportion of cases in which A is wrong and B is correct.

From Brill's paper:

$$Comp(A, B) = 1 - (\frac{common\ errors}{errors\ only\ in\ A}) \quad (1)$$

In this case, $Comp(A, B)$ shows the proportion of errors in A that could be corrected by using B's responses. Another way of looking at it is that $Comp(A, B)$ shows the maximum improvement that a B can make over A; a value of 1 means that B can correct all of A's errors.

Complementarity works well in situations where a label must be assigned to every entry. For example, in part-of-speech tagging there is no argument over which tokens should be labelled: every token gets a label. This is a classical classification scenario, where for a set of instances I, each instance $i \in I$ is assigned a class label from inventory C. The could also be something like assigning a spatial relation, or determining the language of a text.

When combining approaches, it is useful to know how different their performances are. The idea of combining classifiers into *ensembles* is not new to NLP or in general: there is considerable research in supervised (Zhou et al., 2005) semi-supervised (Dasgupta and Ng, 2009) and unsupervised (Zimek et al., 2013) ensembles. Complementarity

has uses in building ensembles: intuitively, such constructions require diversity to function, and this intuition is borne out in practice (Zhou et al., 2005; Uzuner et al., 2011). Regardless of how voting is conducted, at least one contributing system should get the right answer, and complementarity indicates when this happens.
Complementarity has three key behaviours. Namely:

- High disagreement is a strong indicator of error. This is a byproduct of how approaches to many tasks that use similar information will achieve similar performance.
- Complementarity is additive. As more systems are added and the performance goes of the hypothetically best possible combination rises, the complementary rate of the next system will go down monotonically (see also Brill and Wu (1998), Section 2).
- As the amount of information common to all approaches increases (e.g. training data in a statistical learning context), complementarity reduces (Banko and Brill, 2001)

Now we have a measure for comparing two systems' outputs. This form of complementarity works best in situations where the class label distribution is not heavily skewed, and where every instance is of equal interest.

3. Precision and Recall

Precision and recall are well-suited to evaluating problems where the goal is to find a set of items from a larger set of items. In NLP, this can correspond to finding certain linguistic phenomena in a corpus. For example, one may want to search for named entities in a single news article, or identify predominantly French documents in a multilingual corpus. This is in contrast to situations where one assigned a label to every item in a set, such as part-of-speech tagging, where all tokens receive a PoS tag. Phenomena could be things like named entities (phrases referring to things like places, organisations, people, products); times; events; job titles; words connecting pairs of ideas; and so on.
A factor common to most entity recognition scenarios is that the interesting entity counts for only a small proportion of the words examined. This makes the evaluation sets fairly skewed. Indeed, the imbalance of entities vs. non-entities in named entity recognition rewards machine learning approaches that recognise and take advantage of class skew (Li et al., 2009).
Entity recognition could be thought of as a binary classification task. For each token in a document, it is either part of the required type of entity or not. As a result, the majority of words are marked as negative, and the remaining minority as positive – forming part of an entity chunk. Using conventional classifier accuracy is not so helpful in this instance; getting just a small percentage of class labels wrong can have a large effect on entity recognition. Indeed, labelling every token as out-of-entity would be useless, but give a high classifier accuracy. In these scenarios, the very large class of negatives is not so interesting, and getting a true negative is typically unremarkable; a simple most-common-class approach, which assigns the most frequent label in the dataset to every instance it encounters, will achieve perfect performance on the negative class and generate no "false positives".
For this reason, entity recognition performance is commonly measured using precision and recall (Jardine and van Rijsbergen, 1971). These measure the quality and quantity of entities found by the recognition approach.
Precision represents the proportion of items – in this case, entities – that the system returns which are accurately correct. It rewards careful selection, and punishes over-zealous systems that return too many results: to achieve high precision, one should discard anything that might not be correct. False positives – spurious entities – decrease precision. It is defined as::

$$P = \frac{|true\ positives|}{|true\ positives| + |false\ positives|} \quad (2)$$

Recall indicates how much of all items that should have been found, were found. This metric rewards comprehensiveness: to achieve high recall, it is better to include entities that one is uncertain about. False negatives – missed entities – lead to low recall. It balances out precision. Recall is defined as:

$$R = \frac{|true\ positives|}{|true\ positives| + |false\ negatives|} \quad (3)$$

It is possible to "cheat" at both these metrics, in that one can design a system with certain behaviours that are of very limited use but achieve high scores. For recall, if a system simply returns everything possible, then by definition is has returned every correct answer. There are no false negatives. Such a system would have recall of 1.0, but contribute little. Slightly harder is the case of cheating precision. If one correct answer can be found, then that is all that needs to be returned. Because no false positives are generated, and the numerator is above zero, so this gives maximum precision 1.0.
Together, these metrics can be balanced out. These extreme situations used to exploit them contrast with each other: returning everything gives only a baseline precision, and returning just one thing typically gives a very low recall measure. Therefore, it is common practice to combine precision and recall with a weighted harmonic mean, as an F-score (Van Rijsbergen, 1974).

$$F_\beta = (1 + \beta^2)\frac{PR}{\beta^2 P + R} \quad (4)$$

In this equation, coefficient β determines the balance between precision and recall, with high values favouring recall.[1] This is a harmonic weighted mean of precision and recall.
When evaluating using these metrics, focus is given to true positives, false positives and false negatives. However, no attention is given to the large and uninteresting true negative group, which is also the majority class. The result is that one is able to appropriately evaluate differences in classifier performance that, when reported just as part of

[1] Typically F-score is used with $\beta = 1$, c.f. its sometimes being called "F1 score."

overall per-instance classification accuracy, would be small and perhaps insignificant.

As with any lossy operation, the blending of precision and recall into a single F-score has disadvantages. Chiefly, one is unable to distinguish low-recall from low-precision systems.

In feature ablation, a machine learning analysis technique commonly used in NLP, groups of features are removed and the system re-run over the same training and evaluation data.The goal of this exercise is to assess the impact each group of features has on the overall evaluation score by monitoring the change in precision, recall or F-score. This analysis method is somewhat opaque: the same scores can be generated by markedly different behaviours. In ablation, it is important to see the differences in system behaviours. If one feature group's removal keeps overall performance similar, but selects completely different entities, this indicates a potentially interesting analysis candidate. However, this is not possible using the conventional measures.

Entity recognition is typically difficult, and it is often useful to compare the performance of different entity recognition systems. However, comparing F-score differences doesn't tell us how different the mistakes made by each classifier are, and therefore sheds only minimal light on how helpful classifier combination might be.

4. Complementarity Precision and Recall

We propose a set of metrics that measure complementarity over skewed data. They are based on precision and recall, thus using conventional solutions to evaluation problems that come from the typical skew toward negatives. However, they also take into account complementarity, giving more insight into differences than F-score, precision or recall. We call these *complementary precision*, *complementary recall* and *complementary F-score*.

Given key set Y, and candidate labelling sets B and A, precision and recall can help answer questions in terms of spurious and missed entities.

- What errors does A make?
- What errors does B make?
- What errors made by A are not made by B?
- What errors do both make?

Based on these questions, we can determine F-score in the conventional sense. In the context of complementarity, other sets of data can be examined.

- Shared errors. missed, spurious
- Shared accurate items.
- How much is likely to be gained from combining the two: recall by taking the union (reducing missed, increasing spurious), precision by taking the intersection (reducing spurious, increasing missed).
- For disagreements, such as partial results where only one system finds the items, we can also see: spurious in one, missed in one

4.1. Complementary P, R and F

We define four subsets: two of the reference label set and two of a candidate label set. Firstly, we have sets $positive$ and $negative$, where $positive$ corresponds to instances labelled as belonging to the sought-after class – our target entities – and $negative$ the remainder. In addition, for the candidate labeling, we have sets $correct$ for correctly-labeled instances and $wrong$ for wrongly-labeled instances. This gives us the properties:

$$|correct \cap wrong| = 0 \tag{5}$$

$$|positive \cap negative| = 0 \tag{6}$$

$$|correct| + |wrong| = |positive| + |negative| \tag{7}$$

We write these groups as a subscript. For example, $A_{correct}$ is the correctly-labeled instances (that is, the union of true negatives and true positives) returned by system A.

We must also define complementary rate. As per (Brill and Wu, 1998), this is the maximum improvement that one candidate labeling can offer over another. That is, $Comp(A, B)$ indicates the maximum proportional error reduction that B offers over A.

$$Comp(A, B) = \begin{cases} |B_{wrong} = 0|, & 1 \\ otherwise, & 1 - \frac{|A_{wrong} \cap B_{wrong}|}{|A_{wrong}|} \end{cases} \tag{8}$$

If B returns a perfect result, $Comp(A, B) = 1$.

Complementary rate is agnostic to the set of labels involved, and so can also be applied beyond binary classification. Therefore, the $positive$ and $negative$ groups do not feature in its definition. Rather, it just provides the proportion of instances mislabeled by A that would be correctly labeled by B. Note that this is a non-symmetric measure; $Comp(A, B) \neq Comp(B, A)$.

We next define *complementary recall* and *precision*, using similar notations. *Complementary recall* R_{Comp} is the proportion of interesting (i.e. positive) items missed by A that are not missed by B. It describes the error reduction in false negatives that B offers over A.

$$R_{Comp}(A, B) = 1 - \frac{|B_{wrong} \cap A_{wrong} \cap Y_{positive}|}{|A_{wrong} \cap Y_{positive}|} \tag{9}$$

Complementary precision P_{Comp} is the proportion of uninteresting (i.e. negative) items included in A that are not included by B. It describes the error reduction in false positives that B offers over A.

$$P_{Comp}(A, B) = 1 - \frac{|B_{wrong} \cap A_{wrong} \cap Y_{negative}|}{|A_{wrong} \cap Y_{negative}|} \tag{10}$$

Finally, just as F-score combines precision and recall, so we define complementary F-score:

$$F_{Comp}(A, B) = 2\frac{P_{Comp}(A, B)R_{Comp}(A, B)}{P_{Comp}(A, B) + R_{Comp}(A, B)} \tag{11}$$

$F_{Comp}(A, B)$ describes the overall improvement over A that is possible with B, in terms of F-score, in this scenario. The adaptation for F_β is given in Equation 12.

$$F_{\beta Comp}(A, B) = 1 + \beta \frac{P_{Comp}(A, B) R_{Comp}(A, B)}{\beta^2 P_{Comp}(A, B) + R_{Comp}(A, B)} \quad (12)$$

One application of complementary F-score is for comparison to baseline systems. Such comparisons are very common in NLP research, with comparison relative to a prior competitive approach or heuristic being a prevalent evaluation approach. For example, in the SemEval exercises baseline systems are commonly included. While an absolute improvement over baseline in F-score can indicates better performance, to analyse and explain the finding, it is important to know how different from the baseline a target system is – not only overall, but also in terms of the metrics we are used to; precision, recall and F-score. Further, many systems often achieve similar scores given a single task over the same data. The typical tight cluster of performance values that this leads to is hard to explain, and can be unpacked and investigated by discovering the differences in entity annotation among outputs.

4.2. Interpretation of complementarity measures

The measures P_{Comp} and R_{Comp} can signal certain effects from the combination of system results. These allow one to construct simple ensembles which have altered entity extraction performance.

In general, one can increase recall at the cost of precision by taking the union of multiple approaches' entity selections, or increase precision at the cost of recall by taking the intersection of entity selections. Complementarity can provide extra insight into the performance changes likely to result from taking the union or intersection of result sets.

If R_{Comp} is high, then the selection of entities retrieved is markedly different. Combining systems by taking the union of results is likely to lead to a higher increase in recall than in the general case. Conversely, if R_{Comp} is low, little is likely to be gained by taking the union of system outputs.

If P_{Comp} is high, then the selection of true positive entities differs between systems. This means that taking the intersection of results will lead to a larger drop in overall result set size than other approaches (like voting). If P_{Comp} is low, taking the intersection of results will not lead to a big change in true positive rate. Also, there is a higher chance of maintaining high precision when taking the union of results, as the perturbation in true positives will be low.

5. Example Scenarios

All examples use non-synthetic, real datasets, where one is interested in a minority of cases that are significant in some way or another.

5.1. Named Entity Recognition

Named entity recognition focuses on the identification of rigid designators (Kripke, 1972) in text, typically proper nouns that refer to a particular object. For example, one may want to recognise names of all locations, people and organisations in text. It is a long-standing challenge in NLP with many applications (Nadeau and Sekine, 2007).

System	Precision	Recall	F1
Standard			
ANNIE	68.20	83.72	75.17
OpenNLP	81.45	53.29	64.43
Complementary			
Comp(ANNIE, OpenNLP)	78.15	14.22	24.06
Comp(OpenNLP, ANNIE)	20.00	56.04	31.67

Table 1: Named entity recognition: first standard precision, recall and F-score, and then complementary.

System	Precision	Recall	F1
Standard			
LTG	85.1	82.2	83.6
TRIPS	85.1	85.1	85.1
Complementary			
Comp(LTG, TRIPS)	32.5	6.37	10.7
Comp(TRIPS, TRIOS)	15.5	4.55	7.03

Table 2: Temporal expression annotation: first standard precision, recall and F-score, and then complementary.

For this example, we look look at location names present in the CoNLL 2003 dataset (Tjong Kim Sang and De Meulder, 2003). We compare two approaches based on different technologies: ANNIE, which uses gazetteers and finite state transducers (Cunningham et al., 2002); and Open NLP, which uses statistical learning to perform sequence labelling (Baldridge, 2005). Results are shown in Table 1. The first row of complementary figures are for the maximum improvement that OpenNLP offers over ANNIE, and the second for the maximum improvement ANNIE offers over OpenNLP. From these results, we can see that OpenNLP offers strong precision improvement potential over ANNIE. Also, from the high complementary recall that ANNIE offers over OpenNLP, adding ANNIE's results is likely to offer strong recall improvements. However, from $P_{Comp}(OpenNLP, ANNIE)$ being only 20.00 we can see that ANNIE and OpenNLP select quite different entities, and that precision gains are not likely to be made.

5.2. Temporal Expression Extraction

Temporal expressions are another kind of entity prevalent in text, this time describing a time or period of time. Recognising these entities is difficult and has been the subject of much recent research, including shared challenges. To evaluate performance over these expressions, We use the TempEval-2 dataset (Verhagen et al., 2010). In the challenge based on this dataset, one system was used as a baseline for others: a good scenario for applying complementary precision and recall. We compare a system based on semantic information, LTG (Grover et al., 2010), to an advanced parser-based system, TRIPS (UzZaman and Allen, 2010). Results are given in Table 2.

In this instance, we can see that despite having similar F-scores in the overall evaluation, the systems score well on quite different results. Both can offer the other an increase in precision, with TRIPS offering higher quality results than LTG. However, only minor changes in recall are

possible, suggesting that both systems make similar errors in terms of false negatives.

6. Related Work

Cohen's kappa (Cohen and others, 1960) is often used to measure the difference between responses of raters (be they human or automatic). This provides a method of measuring disagreement among multiple sources when the ground truth is not known.

Derczynski (2013) notes that complementarity between systems performing temporal relation extraction – a resiliently difficult NLP problem – does not indicate particularly deficient systems, but rather provides hints as to which temporal relations are more difficult to label than others.

Dimililer et al. (2009) use a genetic algorithm to select classifiers which form an ensemble for biomedical entity recognition. They discover that complementarity is a preferred property, but do not perform evaluations to measure this specifically in terms of precision and recall.

Ferris et al. (2013) address the general behaviours of complementarity. This work includes a plethora of distance measures applicable in various situations, such as market power and classifier margin distance. Of note is Lesbesgue measure (Lebesgue, 1902), used here for measuring subsets of potentially high-dimensional space.

Powers (2011) investigates the relation between precision, recall, F-score and ROC and a variety of complementary metrics for assessing system performance.

7. Conclusion

This paper discussed precision, recall and F-score in the context of NLP. The limitation of ability to describe differences in result set perturbations was identified. This was followed by a solution – the use of complementarity. Complementarity was then adapted to work with precision and recall, and F-score. The different properties of complementary precision and recall, and interpretation guidelines, were also given. This was then demonstrated in two entity-extraction cases.

We propose the adaptation of complementary precision, recall and F-score in situations where the differences between system outputs is of help in analysis; particularly in shared evaluation tasks, in feature ablation, and in cases where traditional F-score results are very close.

Acknowledgments

This work received funding under the uComp project by EPSRC EP/K017896/1, FWF 1097-N23 and ANR-12-CHRI-0003-03 in the framework of the CHIST-ERA ERA-NET program.

8. Bibliographical References

Baldridge, J. (2005). The OpenNLP project.

Banko, M. and Brill, E. (2001). Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 26–33. Association for Computational Linguistics.

Brill, E. and Wu, J. (1998). Classifier combination for improved lexical disambiguation. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 191–195. Association for Computational Linguistics.

Cohen, J. et al. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. (2002). GATE: an architecture for development of robust HLT applications. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 168–175. Association for Computational Linguistics.

Dasgupta, S. and Ng, V. (2009). Mine the easy, classify the hard: a semi-supervised approach to automatic sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 701–709. Association for Computational Linguistics.

Derczynski, L. (2013). *Determining the Types of Temporal Relations in Discourse*. Ph.D. thesis, University of Sheffield.

Dimililer, N., Varoğlu, E., and Altınçay, H. (2009). Classifier subset selection for biomedical named entity recognition. *Applied Intelligence*, 31(3):267–282.

Ferris, M. C., Mangasarian, O. L., and Pang, J.-S. (2013). *Complementarity: Applications, algorithms and extensions*, volume 50. Springer Science & Business Media.

Grover, C., Tobin, R., Alex, B., and Byrne, K. (2010). Edinburgh-LTG: TempEval-2 system description. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 333–336. Association for Computational Linguistics.

Jardine, N. and van Rijsbergen, C. J. (1971). The use of hierarchic clustering in information retrieval. *Information storage and retrieval*, 7(5):217–240.

Kripke, S. A. (1972). *Naming and necessity*. Springer.

Lebesgue, H. (1902). Intégrale, longueur, aire. *Annali di Matematica Pura ed Applicata (1898-1922)*, 7(1):231–359.

Li, Y., Bontcheva, K., and Cunningham, H. (2009). Adapting SVM for data sparseness and imbalance: A case study in information extraction. *Natural Language Engineering*, 15(2):241–271.

Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2(1):37–63.

Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics.

Uzuner, Ö., South, B. R., Shen, S., and DuVall, S. L. (2011). 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556.

UzZaman, N. and Allen, J. F. (2010). TRIPS and TRIOS system for TempEval-2: Extracting temporal information from text. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 276–283. Association for Computational Linguistics.

Van Rijsbergen, C. (1974). Foundation of evaluation. *Journal of Documentation*, 30(4):365–373.

Verhagen, M., Sauri, R., Caselli, T., and Pustejovsky, J. (2010). SemEval-2010 task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62. Association for Computational Linguistics.

Zhou, G., Shen, D., Zhang, J., Su, J., and Tan, S. (2005). Recognition of protein/gene names from text using an ensemble of classifiers. *BMC bioinformatics*, 6(Suppl 1):S7.

Zimek, A., Gaudet, M., Campello, R. J., and Sander, J. (2013). Subsampling for efficient and effective unsupervised outlier detection ensembles. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 428–436. ACM.