A critical survey on measuring success in rank-based keyword assignment to documents

Natalie Schluter
Center for Language Technology, University of Copenhagen, Copenhagen, Denmark
natschluter@hum.ku.dk

Abstract. Evaluation approaches for unsupervised rank-based keyword assignment are nearly as numerous as are the existing systems. The prolific production of each newly used metric (or metric twist) seems to stem from general dissatisfaction with the previous one and the source of that dissatisfaction has not previously been discussed in the literature. The difficulty may stem from a poor specification of the keyword assignment task in view of the rank-based approach. With a more complete specification of this task, we aim to show why the previous evaluation metrics fail to satisfy researchers' goals to distinguish and detect good rank-based keyword assignment systems. We put forward a characterisation of an ideal evaluation metric, and discuss the consistency of the evaluation metrics with this ideal, finding that the average standard normalised cumulative gain metric is most consistent with this ideal.

1 Introduction

Automatic keyword assignment is concerned with assigning documents their representative keywords, either through extracting directly from the text (keyword extraction), or some other process (keyword generation). The task plays an important role in important approaches to, for example, document indexing (for search engines), summarisation, clustering, and classification. Research in data-driven methods for automatically finding keywords for documents has been both classifier-based and rank-based. In recent years, the rank-based systems have come to dominate and appear to form the state-of-the-art under all forms of the keyword assignment task. However, with the move towards rank-based systems, there is a general sense that previously adopted techniques of system evaluation (especially set-based) are inadequate, as testified by the fact that virtually every new publication on the topic introduces some new evaluation metric or metric *twist*. In a nutshell, the problem stems from the added dimension of *ranking*, where the order of items in the list is meaningful, unlike classification systems.

Following the definition of rank-based keyword assignment systems and a discussion of the specification of the keyword assignment task in view of this definition (Section 2), we present a critical survey of the evaluation approaches to this task adopted in the past and attempt to highlight some crucial weaknesses with respect to the rank-based system approach (Section 3). We finish by arguing that the currently most consistent approach to evaluation in this context makes use of the standard Normalised Discounted Cumulative Gain (NDCG) metric, which is mathematically proven to distinguish between ranking systems where one system is substantially better than another (Section 3.3).

2 The definition of rank-based keyword assignment systems

We define rank-based keyword assignment as the following (Cf. (Wang *et al.*, 2013)).

Definition 1. *Given a set of candidate keywords $\mathcal{X} = \{x_1, \ldots, x_m\}$ for a document D and a set $\mathcal{Y}$ of degrees of relevancy, a* ***rank-based keyword assignment system*** *$f : \mathcal{X} \to \mathcal{Y}$ generates a score $f(x) \in \mathcal{Y}$, according to which the m keywords in $\mathcal{X}$ can be organised, resulting in the returned list $x^f_{i_1}, \ldots, x^f_{i_m}$ ($i_j \in [m]$ for all $j \in [m]$ and $i_{j_1} \neq i_{j_2}$ if $j_1 \neq j_2$), which satisfies $f\left(x^f_{i_1}\right) \geq \cdots \geq f\left(x^f_{i_m}\right)$.*

The set $\mathcal{Y}$ can, for instance, be the set of real numbers or the interval $[0, 1]$. It can also be finite. If we set $\mathcal{Y} = \{0, 1\}$, then we see that a classification-based system can be viewed as a simple type of rank-based keyword assignment system. Thus any ranking-system specific evaluation metric can be used for their evaluation also (though indeed this introduces excessive complexity if there is no specific comparison with more complex ranking-based systems).

The original keyword assignment problem does not ask for a *ranking* of keyword candidates, but a set of *correct* keywords. Keywords are short representations of documents ; and as a set they come in small numbers. One could think of this small set size as part of their definition. So, researchers of rank-based systems have generally resorted to returning the top n items of the ranked list. The question of how many keywords to return then arises, since rank-based systems have only organised the candidate keywords into a list (in which all possibilities appear).

Cutting the returned keyword list from a rank-based system off at n, simply because n is the number of positives (correct keyword candidates), seems to be too harsh : a cut-off at $n = 5$, when, say, the sixth and seventh keywords in the list are correct is not the full story. At the same time, a cut-off at $n = 78$ seems far too generous (to at least the recall score) and contrary to the definition of a keyword. Indeed, there seems to be some small upper limit on the size of a keyword set, though it is not clear what this is ; call this observation (O1).

Moreover, we need to be conscious of how much tolerance for error a user or down-stream application receiving the keyword set has. Perhaps precision of at least $\frac{1}{3}$ is tolerable, but $\frac{1}{5}$ becomes fairly useless. Obviously, the higher density and quantity of true positives, the better, but this bound is better determined by the down-stream application. There exists some reasonably low bound on the error tolerance of keyword assignment systems for down-stream applications, though it is not clear what this is ; call this observation (O2).

With the ranking approach to keyword assignment, as Liu et al. Liu *et al.* (2010) note, the ranking order of extracted keyphrases is an important indicator for method preference. We argue that this ranking order in Definition 1 completely specifies the task and accounts for (O1) and (O2) if we observe that producing correct keywords lower down the list in the ranking should be not as important as producing correct keywords high up in the list, as some sort of quality control. We are not sure what the keyword set size is, but we are aware that it should be small by (O1), therefore a decaying "reward" for finding a correct keyword as we move down the ranked list can account for this. Moreover, by (O2), we are aware that down-stream applications could probably afford some density of errors, but that this density should be small ; since the set of correct (gold) keywords is small, as we move down the ranked list, the density of errors probably increases. Once again a decaying "reward" for finding a correct keyword as we move down the ranked list can account for this. Call this observation (O3).

3 Previous system evaluation and their adequacy for rank-based system evaluation

Four general categories of approaches to rank-based keyword assignment system evaluation have been adopted in the past, each category differing in its selection of the parameter n : (1) low choice(s) of n, (2) choice of n as a function of document length, (3) considering all values of n as equal, and (4) oracle choice of n (i.e., the choice of n which maximises the evaluation metric). We discuss these now, in turn.

3.1 Selecting a strong list cut-off n : precision, recall and f-score

The majority of systems have been evaluated using the popular measures of precision (P), recall (R) and f-score (F_1), where $P := \frac{\text{correctly returned keywords}}{\text{returned keywords}}$, $R := \frac{\text{correctly returned keywords}}{\text{all correct keywords}}$, and $F_1 := 2 \cdot \frac{P \cdot R}{P+R}$.

3.2 Keyword set size at a constant cut-off

When using these set-based evaluation measures, researchers typically choose a set size n, and true to the definition of a keyword (keyword set), this parameter is usually chosen to be small.

(Wan & Xiao, 2008) and (Wan *et al.*, 2007) evaluate systems using precision, recall and f-score at $n = 10$ explaining that 10 is the limit, because the guidelines they set for the manual annotation of keywords of the DUC2001 documents gave a limit of 10 keywords. Semeval 2010 task 5 organisers evaluated submitted systems using precision, recall, and f-score, with $n \in \{5, 10, 15\}$ (Kim *et al.*, 2010). (Liu *et al.*, 2010) present precision, recall and f-scores for specific n values selected with respect to the dataset : $n = 5$ for the Inspec dataset and $n = 10$ for the DUC2001 dataset. [1] (Litvak

1. We note that the mean number of keywords in the Inspec training set documents is 9.788 with standard deviation of 4.877. Also, this number was found to be normally distributed with high probability ($K^2 = 127.384, p = 0.0$). Therefore, (Liu *et al.*, 2010)'s value for $n = 5$ cannot be motivated

& Last, 2008) are generous with the smallness boundary (perhaps unrealistically), reporting precision, recall and f-scores for $n \in \{10, 20, 30, 40\}$.

The decision of (Wan & Xiao, 2008) and (Wan *et al.*, 2007) to evaluate with n as the number of positives may be too harsh for a rank-based system, because it allows no error tolerance, which goes against (O2). Moreover, f-scores can be highly chaotic when n is so low.

Consider the hypothetical systems in Table 1, which shows the ranked keyword lists of each of the systems, where 0 is a false positive and 1 is a true positive, and there are a total of 7 positives in the data. The highest f-score is achieved at $n = 8$; however evaluating, as in the Semeval 2010 task 5 at $n \in \{5, 10, 15\}$ doesn't provided any evidence of this. In fact, at $n = 5$, Systems 1 and 2 are tied, and at $n \in \{10, 15\}$, all three systems have the same f-score. However, we can observe some behaviour of the three systems which clearly demarcates System 2 as generally superior given these results, since it finds the positives *earlier* than the two other systems. Unfortunately, f-score at arbitrary cut-offs cannot account for this.

	System 1		**System 2**		**System 3**	
n	kw	f-score	kw	f-score	kw	f-score
1	0	0	1	0.29	0	0
2	0	0	1	0.5	0	0
3	1	0.22	1	0.67	0	0
4	1	0.4	0	0.6	0	0
5	1	0.55	0	0.55	1	0.18
6	0	0.5	1	0.67	1	0.33
7	0	0.46	1	0.77	1	0.46
8	1	0.57	1	**0.85**	1	0.57
9	1	0.67	0	0.8	1	0.67
10	1	0.75	0	0.75	1	0.75
11	0	0.71	1	0.82	0	0.71
12	0	0.67	0	0.78	0	0.67
13	0	0.63	0	0.74	0	0.63
14	1	0.7	0	0.7	0	0.6
15	0	0.67	0	0.67	1	0.67

TABLE 1 – F-scores of hypothetical systems at various levels of n. kw stands for keyword. A value 0 in this column indicates a false positive at the corresponding rank level n, and a value 1 indicates a true positive. The total number of positives is set at 7.

3.2.1 Keyword set size as some fraction of document size

One approach to the manner in which n is chosen for evaluation using precision, recall and f-score is to let n be some fraction of the length of the document. (Mihalcea & Tarau, 2004) used the knowledge of the relatively short length N of the documents (which were abstracts from the Inspec corpus), and set $n := \frac{1}{3} \cdot N$. Clearly the approach of taking first one-third of the returned ranked list does not work for longer sized documents, where n could end up in the 1000s. But, that does not mean that N could not be used to guide the selection of n.

Abstraction made of the problem of the harsh cut-off outlined in the previous section, an additional problem remains. This approach assumes that there is some correlation between n and document size N. We tested this assumption for the training set of Inspec corpus. This training set consists of 1000 scientific abstracts, which form the document set, and as in all previous uses of the corpus (to our knowledge) we considered the set of keywords for abstracts, which were designated as *uncontrolled* (Hulth, 2003). We carried out the D'Agostino-Pearson test for normality on document length, which showed that N is highly likely to be normally distributed ($K^2 = 114.565, p = 0.0$). The number of keywords for document n was also found to be normally distributed by the same test for normality ($K^2 = 127.384, p = 0.0$). We then calculated the Pearson correlation of N and n, however, to discover that there is in fact no correlation between these variables ($R = 0.060$). As such, this method for choosing some appropriate n does not replace the previous one.

On a related note, (Liu *et al.*, 2009) adopt a similar approach to "parameter n problem" in keyword assignment system, reporting precision, recall and f-score for different top fractions of the returned keyword list size : $n := r \cdot m$, where, we recall, m is size of the complete returned ranked list of candidate keywords, and $r \in \left\{\frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{4}{5}\right\}$. It is not possible to carry out regression and significance tests for such an approach, since different systems return different candidate lists. However, r is also a choice.

by the Inspec data.

3.2.2 Oracle selection of n (best parameter evaluation)

(Hasan & Ng, 2010) consider several different datasets (DUC2001, Inspec, NUS, and ICSI) comparing their re-implementations of four previously published systems with a simple baseline, giving oracle parameter settings–n for the specific system and dataset that maximises the f-score ; n varies widely from one system to then next (as well as from one dataset to the next), and from $n = 9$ through to $n = 190$, to the length of the candidate list. This type of evaluation may seem like cheating in the unsupervised setting. And though we do not advocate solely reporting results for oracle parameter settings, we do note that such results are very informative in the following sense. They reveal how fast systems reach their own optimality. If n is large at a systems best f-score, then this is generally a failure of the system. However, if n is fairly small, this can be seen as a success of the system. Moreover, if a system outperforms others at a reasonably low n, it seem fair to say that the system performs best. We believe this is what was intended to be shown by the ROC curves and precision-recall curves of the following section (Section 3.2.3) ; though these attempts, we hope to show, are problematic.

3.2.3 Curves and summarising over all n

With arbitrary or informed selection of n becoming increasingly unsatisfactory, researchers have attempted to avoid this selection altogether, by sketching the curve over all values of n, and drawing conclusions from this curve by means of either taking the area under the curve (AUC) to obtain a single numerical representation, or discussing how one curve *dominates* another.

We hope to show that neither of these strategies is really appropriate for the evaluation of keyword assignment systems. In fact, the former strategy can already be refuted by use of the definition of a keyword (keyword set). The strategy of taking the area under a curve over all values of n treats these values as equal. However, values of metrics at large n should at least be less important than values of metrics at small n (by (O3)). Still, we discuss further faults of the two types of curves previously adopted for rank-based keyword assignment system evaluation : ROC curves and precision-recall curves. [2]

ROC curves. (Litvak & Last, 2008) calculate the AUC of the (average) ROC curve as a means of evaluation.

The ROC curve of a binary classifier plots the true positive rate (TPR) (on the y-axis) against the false positive rate (FPR) (on the x-axis) at incremental levels of n, where $TPR := R$ and $FPR := \frac{\text{false positives}}{\text{negatives}}$.

When using such an evaluation approach for rank-based systems, an immediate problem is therefore the question of true negatives. It is not obvious what can be used as a true negative of a rank-based system for keyword assignment systems. However, ignoring this fact, other problems with the AUC ROC metric persist.

There are some important weaknesses about this measure that are vital to understand for classifiers evaluation in general. [3] Specifically for the case of keyword assignment systems, in addition to the weakness mentioned above for all curve-based metrics, a critical short-coming is the metrics proven inability to always determine the best system when ROC curves cross (which is likely to happen when systems have performances worth comparing) (Hand, 2009; Lobo *et al.*, 2008).

Precision-recall curves. A precision-recall curve plots recall on the x-axis and precision on the y-axis. (Hasan & Ng, 2010) and (Liu *et al.*, 2010) plot precision-recall curves and discuss curve dominance. However, (Davis & Goadrich, 2006) mathematically prove that an algorithm dominates in precision-recall space if and only if it dominates in ROC space. Therefore such an evaluation method is problematic for the reasons already outlined above.

3.3 Standard Normalised Discounted Cumulative Gain

The metric we propose in this paper, (standard) Normalised Discounted Cumulative Gain (NDCG), is already widely used in information retrieval and machine learning research on ranking. It is defined as follows for the keyword assignment task.

2. In fact there is a third curve in the literature. (Litvak & Last, 2008) provide a graph of cumulative AUC for the average precision, with respect to $n \in [1, 589]$. But this seems simply to be a way to plot precision as a smooth curve, which can really only be used to detect an optimum precision point. Therefore, we do not discuss this type of curve here, but refer to Section 3.1.

3. See (Hand, 2009; Lobo *et al.*, 2008) for a complete discussion.

Definition 2. *Let f be a keyword ranking function and $S_i = \{x_{i,1}, \ldots, x_{i,m_i}\}$ be the dataset of keyword candidates for the document D_i, with $|S_i| = m_i$. The **Discounted Cumulative Gain** (DCG) of f on S_i (yielding the ranked list $x^f_{j1}, \ldots, x^f_{jm_i}$) is defined as*

$$DCG(f, S_i) := \sum_{t=1}^{m_i} \frac{\mathbb{I}\{x^f_{jt} \textit{ is a correct keyword}\}}{\log(1+t)}.$$

*The **Ideal DCG** of f on S_i is defined as*

$$IDCG(f, S_i) = \max_{f'} DCG(f', S_i).$$

*The **NDCG** of f on S_i is defined as*

$$NDCG(f, S_i) := \frac{DCG(f, S_i)}{IDCG(f, S_i)}.$$

For system evaluation, one would present the average NDCG over all documents.

We observe from the definition that the evaluation metric is in line with (O3), having a decaying reward of success with respect to rank : $\frac{1}{\log(1+\text{rank})}$. Moreover, an important result on standard NDCG is that every two substantially different [4] ranking functions are *consistently distinguishable* [5] by standard NDCG (Wang *et al.*, 2013). [6] This makes the metric attractive in and of itself.

As illustration, let us consider again our hypothetical systems from Table 1, which were not always distinguishable using the precision, recall and f-score evaluations at $n \in \{5, 10, 15\}$. Their NDCG scores are given in Table 2. We see that the NDCG metric is reflective of our observations on the system performance : System 2 is best, followed by System 1, and last System 3. For further analysis, we can look at the best-parameter f-scores. We see that System 2 achieves its optimal with $n = 8$ (and precision $6/8$ and recall $6/7$), which further assures us that the system is also preforming at its optimal with a good size n. We admit that this is a toy example, but it is only meant for illustration of the concepts and discussion of this paper and not as their proof. For a proof, the reader is referred to, for example, (Wang *et al.*, 2013).

System 1	System 2	System 3
0.681	0.939	0.613

TABLE 2 – NDCG scores for the three hypothetical systems.

On a side note, Liu et al. (Liu *et al.*, 2010) also introduce two new metrics for keyword evaluations : *mean reciprocal rank* and *binary preference measure*. These latter two metrics are meant to account for the ranking order of extracted keyphrases. Unfortunately, for the binary preference measure, the same n parameter must be chosen by the evaluator and for the mean reciprocal rank, only the rank of the first positive keyword in the ranked lists is accounted for. Therefore, we do not consider these as appropriate measures for keyword assignment.

4 Conclusion

Evaluation metrics should fit the task at hand. We hope to have shed some light on how the keyword assignment task should be re-specified under the rank-based approach. In doing so, we have been able explain some important weaknesses of the numerous pre-existing approaches to keyword assignment system evaluation, and motivate (and illustrate) an ideal evaluation metric : average standard NDCG.

Références

DAVIS J. & GOADRICH M. (2006). The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, p. 233–240, New York, NY, USA : ACM.

4. Roughly "substantially different" means "not almost always the same" (Wang *et al.*, 2013).

5. A function $neg : \mathbb{R} \to \mathbb{R}$ is negligeable if for all c, $neg(N) < N^{-c}$, for sufficiently large N. So, roughly, two ranking systems are"consistently distinguishable" by some metric if there exists some negligible function $neg(N)$, that shows preference for one system over the other, with probability $1 - neg(N)$ when keyword candidate lists are at least as big as N (Wang *et al.*, 2013).

6. See Wang et al. (Wang *et al.*, 2013) for details.

HAND D. J. (2009). Measuring classifier performance : A coherent alternative to the area under the roc curve. *Machine Learning*, **77**, 145–151.

HASAN K. S. & NG V. (2010). Conundrums in unsupervised keyphrase extraction : making sense of the state-of-the-art. In *Proceedings of the 23rd International Conference on Computational Linguistics : Posters*, COLING '10, p. 365–373, Stroudsburg, PA, USA : Association for Computational Linguistics.

HULTH A. (2003). Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, EMNLP '03, p. 216–223, Stroudsburg, PA, USA : Association for Computational Linguistics.

KIM S. N., MEDELYAN O., KAN M.-Y. & BALDWIN T. (2010). Semeval-2010 task 5 : Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Worskhop on Semantic Evaluation, ACL 2010*, p. 21–26, Uppsala, Sweden.

LITVAK M. & LAST M. (2008). Graph-based keyword extraction for single-document summarization. In *Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization*, MMIES '08, p. 17–24, Stroudsburg, PA, USA : Association for Computational Linguistics.

LIU Z., HUANG W., ZHENG Y. & SUN M. (2010). Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, p. 366–376, Stroudsburg, PA, USA : Association for Computational Linguistics.

LIU Z., LI P., ZHENG Y. & SUN M. (2009). Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing : Volume 1 - Volume 1*, EMNLP '09, p. 257–266, Stroudsburg, PA, USA : Association for Computational Linguistics.

LOBO J. M., JIMENEZ-VALVERDE A. & R. R. (2008). Auc : a misleading measure of the performance of predictive distributive models. *Global Ecol. Biogeogr.*, **17**, 145–151.

MIHALCEA R. & TARAU P. (2004). Textrank : Bringing order into text. In *Proceedings of EMNLP*, p. 404–411.

WAN X. & XIAO J. (2008). Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the 23rd national conference on Artificial intelligence - Volume 2*, AAAI'08, p. 855–860 : AAAI Press.

WAN X., YANG J. & XIAO J. (2007). Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In *ACL*.

WANG Y., WANG L., LI Y., HE D. & LIU T.-Y. (2013). A theoretical analysis of ndcg type ranking measures. In *Proceedings of COLT*, p. 25–54.