An intelligent WWW agent for similarity-based searching

AN INTELLIGENT WWW AGENT FOR SIMILARITY-BASED SEARCHING

Tony G. Rose[1] and Peter J. Wyard

Abstract

This paper describes the development of a WWW agent that uses similarity-based methods to search the Internet. The Internet Information Agent (IIA) works by analysing a sample of the type of text that is known to be of interest to the user. It then extracts a number of linguistic features and stores these as a feature vector that is used to describe the content of the document. This data is then used as input to a range of similarity metrics that allow the agent to compare new texts with the original and thereby acquire "more of the same". The agent's strengths lie in its use of a range of similarity metrics that are known to perform well over a wide variety of input. The agent has been tested across a range of input data and evaluated against a number of criteria. The results of this evaluation are described and the prospects for the ongoing development of the agent are discussed.

1.0 Background

The basic actions involved in a typical Information Retrieval (IR) application are: (1) formulation of the query, (2) comparison of the query with the documents in the database, and (3) presentation of results to the user. Since the IIA is ultimately an IR application, it too performs these three basic operations. However, its function involves some distinct characteristics that reflect its working environment and the presence of certain novel features:

1. The document database is the Internet, and this requires specific treatment. For example, the IIA must observe the Robot Exclusion Protocol, be able to cope with time-outs, empty documents, etc.;
2. Since the IIA's mode of operation is to "learn" from a sample of "reference" text presented by the user, it does not ask the user to formulate a separate query. However, it does process the original (or "reference") sample using methods that make it analogous to a query;
3. The agent uses a novel control algorithm that allows it to adjust its own weightings to suit the particular sample of text used as reference material.

The action of the IIA can therefore be summarised by the following three operations: (1) analysing the original or "reference" text, (2) searching the Web and applying the appropriate similarity metric(s), and (3) presenting the results to the user. Of these three, the main activity (and the one by which most competing systems are judged) is the choice and application of the appropriate similarity metrics. Moreover, it is here that the use of novel algorithms and natural language techniques will have most effect.

2.0 The similarity metrics

2.1 The cosine measure

The simplest way for a program to find documents on the Internet is to ask the user to formulate a query and then compare this with the titles that are returned by a HTTP "get" operation. Obviously, a well-written indexing program will save time by performing the second activity off-line, & producing an inverted index to the results that may then be searched on-line via a HTML forms interface. A range of comparison operations

[1] Tony Rose is a senior research scientist at Canon Research Centre Europe, Guildford, GU2 5YF, UK. This work was completed during his previous appointment as a BT Short-Term Research Fellow. Peter Wyard is a senior professional at BT Labs, Martlesham Heath, IP5 7RE, UK.

may then be applied - e.g. the cosine measure (Salton, 1983). This method is particularly popular since it takes into account both term frequency and term scarcity, and handles variations in the length of documents & queries. Indeed, many early Internet robots used only the title since in many cases such simple methods (perhaps augmented by thesaurus information) can be sufficiently effective. However, this approach has some serious shortcomings. By using only the title, and ignoring the content, it requires the authors to choose titles of their various documents with care. Clearly, this requirement is not always met. The IIA adopts a more robust approach by analysing both the title and the content of each document.

Once the IIA has applied the cosine measure to the document title, it begins to analyse the content. The IIA then applies a further three metrics to this data: (a) character-level n-gram analysis, (b) word frequency analysis and (c) word-level n-gram analysis. It is this use of four measures, each exploiting a different aspect of the document, that promises to give the agent both robustness and flexibility. For each metric, a similarity score is calculated and then multiplied by a suitable weighting factor (see Section 3.0).

2.2 Character-level n-gram analysis

An n-gram is a string of symbols. A string of length 2 is known as a bigram, a string of length 3 is known as a trigram, and so on. N-gram distributions can easily be extracted from a text and represented as a rank-ordered frequency list. These lists have the property of being highly dependent on domain, i.e. an n-gram distribution extracted from one subject area will differ significantly from that of another, while n-gram distributions from the same domain tend to share many common features. It is this property that enables character-level n-gram data to be used effectively for text categorisation (Cavnar & Trenkle, 1994). In addition, since any given text will contain more n-grams than words, this method gives robust performance even on very short documents (where word-based techniques would suffer from data sparsity). The distributions can be analysed using a variety of statistical tests; rank correlation (using Spearman's S) is typical. However, other tests may be suitable and studies are currently in progress to determine the best combination of accuracy and reliability.

2.3 Word frequency analysis

As with character-level n-gram analysis, the basic unit of comparison is the frequency list. However, in this case, the frequencies are those of words rather than n-grams. Consequently, a similar range of statistical measures may be applied (Daille, 1995). Work is currently underway to identify a measure which does not suffer from some of the known shortcomings of other well-used metrics, e.g. *chi-square*, (which tends to over-emphasise high contingency values) and *mutual information* (which tends to over-emphasise low contingency values). In addition, the ideal measure should make no assumption of normally distributed data and cope well with varying amounts of information, which is typically the case with data extracted from textual sources.

2.4 Word-level n-gram analysis

The purpose of word-level n-gram analysis is to help address one of the major challenges within information retrieval: the need to perform reliable word sense disambiguation (WSD). For example, a search for the word "engine" would return documents related to many types of engine, i.e. automotive, computational, Internet search engines & so on. The search can be narrowed down using Boolean operators (e.g. "search" AND "engine") but that could still return documents in which someone was "searching" for parts of a car "engine".

A simple way to perform WSD is to first process the reference text by extracting all the word-level n-grams up to a certain length (e.g. 3). When these are compared with those found in the candidate document, a better match will be produced if words are found in their "correct" context, i.e. when "search" and "engine" are found in the bigram "search engine" rather than found separately. The key to successfully implementing this process is to find a method by which the n-grams can efficiently be extracted and then applied in such a way as a reliable score is produced even if the n-grams provide only a partial (rather than exact) match. Such a method has been identified, and is described further in Rose & Wyard (submitted). This method enables the IIA to perform an elementary form of WSD that can help discriminate between documents that appear similar at the word and character n-gram level, but are actually from different domains and are therefore using the words in different senses.

3.0 Analysis of the reference text

Before the IIA begins its search, it performs an analysis of the reference document. Firstly, it performs a word count of the reference text. Secondly, it performs a homogeneity test, which measures the amount of linguistic variation in the reference text. It does this by randomly allocating sentences from the reference text to one of two "sub-texts", and then comparing those two sub-texts with each other, using the similarity metrics described earlier.

This process is a crucial step. Since the IIA traverses the Internet trying to find "more of the same", its performance will depend almost entirely on the effectiveness of its similarity measures. However, the result of a similarity measure between two texts is only meaningful *if the two texts are consistent within themselves* (Kilgarriff, 1996). The greater the degree of internal consistency, the more confidence there will be in the results of a similarity test.

The results of these tests may then be used to optimise performance. For example, if the reference document is a single page, it is unlikely that there are sufficient word-level n-grams to constitute a representative sample. This metric should therefore be applied with a very low weighting. Similarly, word frequency data may be sparse, so this metric should possibly be given a lower weighting. Consequently, the majority of the weighting would be reserved for the character-level n-gram analysis, which is less susceptible to the problems caused by sparse data. Conversely, if the reference text is an entire web site (or any other source containing several thousand words) then the weightings should be adjusted to reflect the increasing reliability of the other metrics. The weightings may then be further modified by the results of the homogeneity test to provide an overall measure of confidence in the matching process and the resultant similarity scores. The precise details of this algorithm can be found in Rose & Wyard (submitted).

Before the IIA begins its search, the user is offered the opportunity to override any of the recommended weightings. The agent then reads its search directives from a task file. These directives cover details such as: where to commence searching, how many levels down to search, what boundaries there are on the current search, which directories should be excluded, etc. The search then proceeds in a breadth-first manner, exploring links as it finds them. The action of the IIA can be defined mathematically by saying that it calculates for each document a similarity score S_{doc} according to:

$$S_{doc} = (w_1 * (1\text{-}m1_{doc})) + (w2 * m2_t) + (w3 * m3_{doc}) + (w4 * m4_{doc})$$

where m1 = cosine measure; m2 = character n-gram analysis; m3 = word frequency analysis; m4 = word level n-gram analysis and $m1_{doc}$ to $m4_{doc}$ = the results of applying metrics m1 to m4 to the candidate document. It should be noted that metrics 2 to 4 are actually *dissimilarity* metrics, in that they return a higher score the more dissimilar two documents are. For consistency therefore, m1 (the cosine measure) was recast as a dissimilarity metric by subtracting its value from +1 before multiplying by w1.

4.0 Presentation of results

Once the IIA completes its search (according to the directives given in the task file) it prepares to display the results. Firstly, it sorts the URLs it has visited according to their similarity score, and presents to the user this data as rank ordered list. Secondly it informs the user that the top ranking documents (whose score was above the given similarity threshold) have been saved in a separate file.

5.0 Evaluation

Time constraints have allowed only an initial evaluation of the IIA. Consequently, these results of these studies should be considered only as preliminary findings.

Pilot Study 1

The object of this study was to determine whether the IIA could find a number of documents that had previously been manually identified as relevant, when placed at random within the pages of an unrelated web server. In addition, the rank of those documents and their similarity scores could be compared with those of the highest other returned documents.

For this experiment, the reference (or "training") text was the complete set of pages resident on the BT Language Group web server. (Actually, only 90% was used - the remaining 10% was reserved as test data: see Pilot Study 2). Clearly, the test documents had to be from a different source, but still related to the training material (i.e. concerned with Language Technology). For this reason, the introductory pages from three other centres of language technology research were chosen: UMIST, Durham University and Sheffield University. The documents had then to be placed in a location where the IIA would be sufficiently tested in finding them. This was achieved by were linking them within the "infostructure" of a further (unrelated) BT Web site: the Intelligent Systems Unit server. The overall evaluation procedure was thus:

1. Choose a suitable reference (or "training") text.
2. Select a number of suitable test documents, and link these to the infostructure of an unrelated web server.
3. "Train" the IIA on the reference text, setting the weights accordingly (since the CV of the source was high in this case, the weights were set equally).
4. Let the IIA search the web server for the test pages.
5. Review returned documents.

The results are shown in Table 1. Clearly, they are disappointing, since the average rank of the target is poorer than would be expected by random selection. This highlights the extent to which the weightings need to be correctly tuned for robust performance. In addition, there remains a further question regarding the relationship between document length and similarity. Whilst it is understood that the some metrics should be relatively insensitive to variations in length, it is also evident that longer documents *seem* to produce lower similarity scores. Since the target documents were all of greater length than the average of the others, this may have been a factor. Further investigation of this issue is required.

Page	**Rank**
Sheffield	409
Durham	121
UMIST	397
Average of above	**309**
Median rank (overall)	**302**

Table 1. Ranks of relevant pages found on unrelated server.

Pilot Study 2

This study proceeded as above except that the element of subjectivity in choosing the target document was eliminated. Instead, a set of "pseudo-documents" were created from the reserved 10% of the reference text (the content of the BT Language Group server). These pseudo-documents were constructed to be each around 590 words in length, since this was the average length of the documents used in Study 1. Since the documents were created from test sentences, they had no title. Consequently, the weighting for metric m1 in this experiment was set to zero.

Page	**Rank**
pseudo1	35
pseudo2	117
pseudo3	64
Average of above	72
Median rank (overall)	304

Table 2. Ranks of relevant pages found on unrelated server.

The results are shown in Table 2. Firstly, it is clear that the performance has improved, although by a somewhat modest amount. Since the pseudo documents were created from text that is indisputably from the same source as the training material, a more positive retrieval score would have been expected. Possible reasons for the poor performance again concern the weight setting; the importance of this and the need for further empirical evaluation cannot be underestimated.

In addition, it is necessary to consider the actual source material used as training data. Can it really be said that the contents of a web site constitute a coherent information source? With a multiplicity of authors and document types it is inevitable that there will be considerable variation. This underlines the importance of performing a homogeneity test beforehand, so that the degree of variation may be estimated. More work is needed to identify the best way of using this data to control the automatic weight setting procedure.

6.0 Further Work

Clearly, these experiments provide only a preliminary evaluation. Further rigorous testing is required, using a recognised test set and the familiar metrics of precision and recall. Moreover, the tests above both used an entire web server (or at least 90% of it) as the starting point - what would be the effect of using a single document? How effective would the IIA then be? These are currently the subjects of further study.

A further improvement is to normalise the output of each similarity metric to produce a value between 0 and 1, where 0 represents two identical documents and 1 the case where they are maximally dissimilar. Another extension to the functionality of the agent is to package the script as a CGI-bin program, present a form-based WWW interface to the user and carry out all interaction via the WWW. This would allow users to interactively view the URLs returned by the IIA. It may also be useful to allow the user to submit searches off-line, and present the results as a URL to a page of HTML (notifying the user by email when the page is ready).

6.1 Dynamic weight updating

The performance of the IIA can only benefit from further tuning. There are many sophisticated metrics involved in its operation and considerable evaluation is needed to optimise the performance of all parameters. One way of achieving this is to incorporate some degree of relevance feedback, in which users can interact with the IIA to improve its performance and more tightly focus its searching. This could be achieved by the user specifying which of the returned documents constitute good examples (in which case the IIA can use them as further training material and update its own reference files) and which are particularly bad examples (in which case the agent could use the information to "purge" its reference files of inappropriate n-grams and word frequency data).

For example, suppose two texts have been returned, one that is rated 5/10 by the user and the other 9/10 (this could be implemented using a spin button, slider bar, etc.) The IIA could then increase the weighting of the metric that made the biggest contribution to the similarity score of the second document, and decrease those that made the smallest contribution. The value of this adjustment would be proportional to the size of the difference multiplied by a suitable normalisation factor. In cases where more than two texts have been returned (and scored by the user), the IIA could work iteratively through the list calculating differences and adjusting weightings accordingly.

6.2 Corpus Acquisition

One of the original aims behind the design of the IIA was to help solve a problem that has troubled the speech recognition community for many years: how to acquire sufficient quantities of training data from which to build reliable language models. In a typical speech recognition application there is only a small quantity of task data available, so attempts are made to augment this with data from a more general source. Often the results are unsatisfactory, since the n-grams obtained from a small sample cannot easily be merged with those derived from a large general corpus (Jelinek, 1990). A variety of interpolation schemes have been suggested, but few have met with widespread acceptance.

The IIA helps solve this problem since it is designed to find (and download) samples of text that match the characteristics of the reference sample. The Internet is surely the biggest corpus in existence, so if the methods for efficiently analysing it can be developed then the speech recognition community would benefit greatly. The IIA is currently being evaluated for precisely this purpose (Wyard & Rose, submitted).

7.0 References

1. W. Cavnar, J. Trenkle (1994) "N-Gram-based Text categorisation", Proceedings of the Symposium on Document Analysis and Information Retrieval, Las Vegas, NV.
2. B. Daille (1995) "Combined approach for terminology extraction", UCREL Technical Paper No. 5, Lancaster University.
3. F. Jelinek (1990) "Self-organized language modelling for speech recognition", in A. Waibel & K. Lee (Eds.) "Readings in Speech Recognition", Morgan Kaufmann.
4. A. Kilgarriff (1996) "Which words are particularly characteristic of a text?" in L. Evett & TG Rose (Eds.) "Language Engineering for Document Analysis & Recognition" (AISB Workshop), Sussex, UK.
5. T.G. Rose, P.J. Wyard (submitted) "A similarity-based agent for Internet searching", paper submitted to RIAO'97, Montreal.
6. G. Salton, J. McGill (1983) "Introduction to Modern Information Retrieval", McGraw Hill.
7. P.J. Wyard, T.G. Rose (submitted) "An Internet agent for language model construction", paper submitted to Eurospeech '97.

Printed and published by the IEE, Savoy Place, London WC2R 0BL, UK.