LBSC 878 Oard/**Soergel**

Dagobert Soergel ds52@umail.umd.edu February 26, 1999

An outline of issues in feature assignment (aka indexing)

This outline presents an overall view of the twin problems of feature assignment and matching.

General framework: Feature assignment and matching

The problem of retrieving documents (broadly defined) can be simply stated as follows:

Draw an inference from what is there (bits, pixels, sounds) to the degree or likelihood of the relevance of a document to a user's need. Such inferences can be drawn by a computer system, a person working on behalf of the user, or by the user him/herself. Often there is a pipeline of processes that each carry out some inference steps.

The inference can be done entirely in the matching process, or we can arrange for some inferences to be done beforehand to make the matching process cheaper and faster. When matching speed is not an issue, resources should be allocated to preprocessing and to matching to minimize overall cost at a given level of retrieval performance. But usually matching time is an issue, in which case preprocessing may be required even in the face of higher total cost.

A **preprocessing step** takes some input and draws inferences to assign features that are linked more closely to the final relevance estimate than the input features. Some **examples**:

From bits or pixels to characters

From pixels to shapes: Determine the shapes present in an image (corresponding matching process: Given a desired shape, look for a pixel configuration that is somewhat like the shape)

Segment a text (given in ASCII codes, as graphic images or sound) into words. This is not always trivial, there may be ambiguity. Checking as part of the matching process whether a given word is in the text may work better in cases where there is ambiguity.

Determine phrases in a text.

Map synonyms to a preferred term (corresponding matching process: expand a query term by ORing all desired synonyms together)

Disambiguate homonyms (corresponding matching strategy: The query representation specifies a term in a given meaning; once that term has been located in the text, the matching process examines the context, possibly using rules defined in the query, to determine whether the term in the text has the desired meaning.

Have a list of expected questions and assign to each document the questions for which it is relevant.

Problems with preprocessing to assign features

Feature assignment requires a commitment to the types of features to be used (and sometimes to a list of specific feature values in each type) and to an interpretation of the input data. For example, mapping from synonyms to preferred terms requires a commitment to the preferred terms and to the synonym relationships. Are *highway* and *freeway* synonyms? Or *creek* and *brook*? In query term expansion, the user has complete freedom of what terms to OR and can even assign different weights to different terms.

Problems with not preprocessing to assign features

Example. If we commit to a set of preferred term, each corresponding expression in a text will be mapped. If the feature assignment is done by a person, that person may see many non-lexicalized expressions that should be mapped to a descriptor (*finding data* maps to *information retrieval*. It is much easier to recognize such expressions while reading a text than it is to think of all such expressions in query term expansion. (This assumes that reading the whole text of all documents for every search would lead to prohibitively long matching time.)

Issues in feature assignment

Type of features (connection to knowledge representation)

Orientation in feature assignment: document-oriented vs request-oriented

Developing the feature list vs applying the feature list. In some processes they do go hand in hand (for example clustering results in both a classification of a set of items and the assignment of each item to a class).

Manual vs automated methods of creating the feature list, aka thesaurus building (for subject features). In both: Starting from documents vs starting from requests (query statements)

Manual vs automated methods for assigning features, as outlined below.

Weights in feature assignment

Knowledge used in feature assignment. Complexity of inference used.

Cost as an important issue

Manual methods for feature assignment

Potential for very sophisticated inference. Issues:

Cognitive processes of indexers (descriptive). Knowledge used by indexers.

Rules for indexing (prescriptive)

Support for indexers (example: MedIndex, discussed under frames)

Correctness of indexing (validity) and indexer consistency (reliability)

Automated methods for feature assignment

Document-oriented methods. These methods start from the document and do not consider expected questions, even if mapping into a controlled set of features takes place

Assigning every word in the text. Stop words. Stemming etc.

Phrase detection - collocations come in here

Word sense disambiguation for concept indexing

Simple mapping from words/phrases into a controlled vocabulary

Selecting from all the words/phrases/disambiguated terms in some way. Word-based? Concept-based? Some criteria:

> Frequency of term or concept in the document compared to collection frequency.
>
> Position of term in the document (title, abstract, in introduction or conclusion, in first or last sentence of paragraph)
>
> Marked by a preceding or following cue word or Phrase (*It is important to note that*, *This result is significant because*)
>
> Use of frames to extract important features (as in EDS)

Request-oriented methods. Start with a set of features derived from questions and examine the documents, drawing inferences from "lower-level" features to determine which question-derived feature(s) apply. Examples:

Text categorization: Inferring the class to which a document belongs from document features. Classes could be defined by subject or by document type (e.g. spam email messages) or other criteria. Inference rules learned from a training set of documents for which the class is known. In deriving user profiles automatically there are two classes: relevant and not relevant; the known relevant documents form the training set.

Language identification.

Computing a readability score.

A special problem in automated feature assignment: Word-sense disambiguation

Approaches

For an entire document (as in Lin) vs. specific occurrences (consider "It is our patriotic duty to pay duty")

Based on subject field: Use text categorization method to identify subject field of document, then use knowledge base to determine meaning of the word in the subject field.

Lin's method

Parsing: Part of speech information. In noun phrases (collocations) ambiguity of the constituent words is often removed.

Semantic restriction rules: Relationships between words (only humans *laugh*, only machines are *fixed*, with humans *white* and *black* generally mean race (except if the context suggests a statement about health status), otherwise they mean color, patriotic does not go with duty in the sense of fees, etc.

Semantic nets: Using a knowledge base like WordNet, construct a semantic net of the concepts referred to in the text or a segment of the text (preferable). If a word refers to several concepts (meanings), consider each alternative. The meaning that is most tightly connected in the semantic net is likely to be the correct one.

LBSC 878 Oard/**Soergel**

Dagobert Soergel ds52@umail.umd.edu February 26, 1999

An outline of issues in feature assignment (aka indexing)

This outline presents an overall view of the twin problems of feature assignment and matching.

General framework: Feature assignment and matching

The problem of retrieving documents (broadly defined) can be simply stated as follows:

Draw an inference from what is there (bits, pixels, sounds) to the degree or likelihood of the relevance of a document to a user's need. Such inferences can be drawn by a computer system, a person working on behalf of the user, or by the user him/herself. Often there is a pipeline of processes that each carry out some inference steps.

The inference can be done entirely in the matching process, or we can arrange for some inferences to be done beforehand to make the matching process cheaper and faster. When matching speed is not an issue, resources should be allocated to preprocessing and to matching to minimize overall cost at a given level of retrieval performance. But usually matching time is an issue, in which case preprocessing may be required even in the face of higher total cost.

A **preprocessing step** takes some input and draws inferences to assign features that are linked more closely to the final relevance estimate than the input features. Some **examples**:

From bits or pixels to characters

From pixels to shapes: Determine the shapes present in an image (corresponding matching process: Given a desired shape, look for a pixel configuration that is somewhat like the shape)

Segment a text (given in ASCII codes, as graphic images or sound) into words. This is not always trivial, there may be ambiguity. Checking as part of the matching process whether a given word is in the text may work better in cases where there is ambiguity.

Determine phrases in a text.

Map synonyms to a preferred term (corresponding matching process: expand a query term by ORing all desired synonyms together)

Disambiguate homonyms (corresponding matching strategy: The query representation specifies a term in a given meaning; once that term has been located in the text, the matching process examines the context, possibly using rules defined in the query, to determine whether the term in the text has the desired meaning.

Have a list of expected questions and assign to each document the questions for which it is relevant.

Problems with preprocessing to assign features

Feature assignment requires a commitment to the types of features to be used (and sometimes to a list of specific feature values in each type) and to an interpretation of the input data. For example, mapping from synonyms to preferred terms requires a commitment to the preferred terms and to the synonym relationships. Are *highway* and *freeway* synonyms? Or *creek* and *brook*? In query term expansion, the user has complete freedom of what terms to OR and can even assign different weights to different terms.

Problems with not preprocessing to assign features

Example. If we commit to a set of preferred term, each corresponding expression in a text will be mapped. If the feature assignment is done by a person, that person may see many non-lexicalized expressions that should be mapped to a descriptor (*finding data* maps to *information retrieval*. It is much easier to recognize such expressions while reading a text than it is to think of all such expressions in query term expansion. (This assumes that reading the whole text of all documents for every search would lead to prohibitively long matching time.)

Issues in feature assignment

Type of features (connection to knowledge representation)

Orientation in feature assignment: document-oriented vs request-oriented

Developing the feature list vs applying the feature list. In some processes they do go hand in hand (for example clustering results in both a classification of a set of items and the assignment of each item to a class).

Manual vs automated methods of creating the feature list, aka thesaurus building (for subject features). In both: Starting from documents vs starting from requests (query statements)

Manual vs automated methods for assigning features, as outlined below.

Weights in feature assignment

Knowledge used in feature assignment. Complexity of inference used.

Cost as an important issue

Manual methods for feature assignment

Potential for very sophisticated inference. Issues:

Cognitive processes of indexers (descriptive). Knowledge used by indexers.

Rules for indexing (prescriptive)

Support for indexers (example: MedIndex, discussed under frames)

Correctness of indexing (validity) and indexer consistency (reliability)

Automated methods for feature assignment

Document-oriented methods. These methods start from the document and do not consider expected questions, even if mapping into a controlled set of features takes place

Assigning every word in the text. Stop words. Stemming etc.

Phrase detection - collocations come in here

Word sense disambiguation for concept indexing

Simple mapping from words/phrases into a controlled vocabulary

Selecting from all the words/phrases/disambiguated terms in some way. Word-based? Concept-based? Some criteria:

> Frequency of term or concept in the document compared to collection frequency.
>
> Position of term in the document (title, abstract, in introduction or conclusion, in first or last sentence of paragraph)
>
> Marked by a preceding or following cue word or Phrase (*It is important to note that*, *This result is significant because*)
>
> Use of frames to extract important features (as in EDS)

Request-oriented methods. Start with a set of features derived from questions and examine the documents, drawing inferences from "lower-level" features to determine which question-derived feature(s) apply. Examples:

Text categorization: Inferring the class to which a document belongs from document features. Classes could be defined by subject or by document type (e.g. spam email messages) or other criteria. Inference rules learned from a training set of documents for which the class is known. In deriving user profiles automatically there are two classes: relevant and not relevant; the known relevant documents form the training set.

Language identification.

Computing a readability score.

A special problem in automated feature assignment: Word-sense disambiguation

Approaches

For an entire document (as in Lin) vs. specific occurrences (consider "It is our patriotic duty to pay duty")

Based on subject field: Use text categorization method to identify subject field of document, then use knowledge base to determine meaning of the word in the subject field.

Lin's method

Parsing: Part of speech information. In noun phrases (collocations) ambiguity of the constituent words is often removed.

Semantic restriction rules: Relationships between words (only humans *laugh*, only machines are *fixed*, with humans *white* and *black* generally mean race (except if the context suggests a statement about health status), otherwise they mean color, patriotic does not go with duty in the sense of fees, etc.

Semantic nets: Using a knowledge base like WordNet, construct a semantic net of the concepts referred to in the text or a segment of the text (preferable). If a word refers to several concepts (meanings), consider each alternative. The meaning that is most tightly connected in the semantic net is likely to be the correct one.