Termination Project, Professor Harold Lewis

Fuzzy Methods in Word Sense Disambiguation

Grady Kurpasi

[gkurpasi@hotmail.com](mailto:gkurpasi@hotmail.com) (213) 304-8541

|  |  |  |
| --- | --- | --- |
| A R T I C L E I N F O  3 May 2020  Keywords: Fuzzy Inference  Fuzzy C-Means  Kernel Fuzzy C-Means,  Hyperspace Analogue to Language  WordNet |  | A B S T R A C T  This write up is the term project for SSIE 617 Fuzzy Sets and Fuzzy . It provides an overview of fuzzy methods applied to Word Sense Disambiguation (WSD). Three papers are reviewed “Constructing Fuzzy Relations from WordNet for Sense Disambiguation”, “Word Sense Disambiguation in Hindi Language Using Hyperspace Analog to Language and Fuzzy C-Means Clustering” and “Kernel Fuzzy C-Means Clustering for Word Sense Disambiguation in Biomedical Texts”. These papers investigate the application of Fuzzy Inference, Fuzzy C-Means clustering and Kernel Fuzzy C-Means clustering respectively. |

1. Introduction

This is the final paper for SSIE 617 Fuzzy Set and Fuzzy Logic. It provides an overview of applications of Fuzzy methods employed in Word Sense Disambiguation. It reviews three papers that employ Fuzzy Inference, Fuzzy C-Means Clustering and Kernel Fuzzy C-Means Clustering..

* 1. Introduction & Paper Organization

Section 1 of this paper, (this introduction), will layout the organization of this paper and describe each section of the body.

* 1. Word Sense Disambiguation

Section 2 will discuss the Word Sense Disambiguation problem. Word Sense Disambiguation is the mapping of ‘words’ to underlying ‘meanings’. It is notoriously difficult to do in an automated manner and is a roadblock to both natural language processing and semantic computing. This section will describe Word Sense Disambiguation in detail. It will also discuss some past and current approaches to the problem as well as some of the difficulties faced by researchers in this field

* 1. Fuzzy Inference

Section 3 will describe the paper “Constructing Fuzzy Relations from WordNet for Word Sense Disambiguation” [1]. Christos Diou, Giorgos Katsikatsos and Anastasios Delopoulos describe their knowledge based approach to WSD that relies on WordNet to build and weight candidate senses so that they can employ fuzzy inference to select a most suitable sense.

* 1. Fuzzy C-means clustering

Section 4 examines “Word Sense Disambiguation in Hindi Language Using Hyperspace Analog to Language and Fuzzy C-Means Clustering”. Devendra K. Tayal, Leena Ahuja and Shreya Chhabra describe their application of Fuzzy C-Means clustering to conduct Word Sense disambiguation. Their solution relies on a Hyperspace Analogue to Language (HAL) encoding to conduct clustering and sense selection

* 1. Kernel Fuzzy C-means clustering

Section 5 describes “ Kernel Fuzzy C-Means Clustering for Word Sense Disambiguation in Bio-Medical Texts”. K. Ren and Y.F. Ren attempt to increase the efficacy of Fuzzy C-Means clustering with kernel mapping applied to Bio Medical texts and disambiguation of technical terms and acronyms

* 1. Conclusion

Section 6 will discuss final thoughts on fuzzy methods in WSD and potential future applications for study.

1. Background on the Application

Word Sense Disambiguation is the mapping of ‘words’ to underlying ‘meanings’. It is a notoriously difficult task for both humans and machines and is currently a major road block to advances in natural language processing and semantic computing. WSD is complicated by the complexity of linguistic data, a lack of agreement on the correct inventory of ‘meanings’ as well as by a lack of standardization in research.

WSD has been a long standing aspiration for computer and information scientists and despite these challenges, advances have been made in the last decade with the application of fuzzy methods and deep learning to the WSD problem.

* 1. Introduction to Word Sense Disambiguation

Word Sense Disambiguation involves most, if not all aspects of linguistics. Linguistics itself is a sprawling field of study that has not yet unlocked many of its own core tenets. Nonetheless, linguistics still provides a robust body of work to analyze and support WSD.

In English, the Word Sense Disambiguation problem can be intuitively illustrated by the word ‘board’. This word can represent a skate-board, surf-board or chess-board. It can also represent a group of decision makers or the act of getting on a ship, train or bus. It can mean covering up a window with planks, or checking a hockey player into the side of a rink.

What we intuitively think of as the *word* ‘board’ can have a lot of different meanings that must be inferred from context. In English this provides what appears to be a straight forward entry point to the Word Sense Disambiguation problem.

In English this also provides a window on the true complexity of Word Sense Disambiguation (WSD). WSD is the act of figuring out exactly what is being talked about or said. WSD attempts to associate linguistic tokens, either written or spoken, with one and only one discrete semantic meaning referred to as a sense.

Unfortunately, though this spans a number of problems in natural language processing. For example, what we refer to as the *word* ‘board’ in written language actually represents several words including at least two nouns and one verb. These words are all homographs. The graphemes ‘b-o-a-r-d’ can represent all of them which complicates WSD.

Disambiguating senses of the word board could alternatively be as easy as conducting Part of Speech tagging and knowing that “He sat on the board” could not refer to the verb ‘get on something’. However, it could be as complicated as requiring discourse level memory to know if he sat on a skate/surf/circuit/ or group of directors ‘board’.

Likewise, in spoken language, the utterance /’bɔrd’/ (in the International Phonetic Alphabet) could be one of the ‘b-o-a-r-d’ words or ‘b-o-r-e-d’. ‘Board’ and ‘bored’ are homophones in English, meaning they are uttered almost identically. Disambiguating senses of this utterance from spoken language is similarly difficult: “The /’bɔrd’/ girl” may generally refer to ‘bored’, but it is impossible to rule out ‘board’ definitively without context. Alternatively, phonologic / phonetic qualities of the utterance of the word may assist in disambiguation. E.g. A long drawn out, maybe creaky pronunciation of /’ɔr’/ (‘OR’ sound) might unambiguously represent ‘booorrrred’.

Using a different example, many irregular nouns in English do not undergo morphological changes in pluralization. E.g. “A fish” is spelled and pronounced identically to “those fish” (as well as the verb “to fish”). Disambiguating this requires a knowledge of language specific morphology and morphological irregularities.

In the rich morphology of agglutinative languages, even defining a word can be problematic. Words in many agglutinative languages are effectively unbounded. Entire sentences can consist of a what is technically a single word adorned with a potentially unending array of affixes that add or modify meaning.

Interpreting ‘words’ and associating them with an intended meaning is the goal of both human and machine word sense disambiguation.

* 1. Linguistic Word Sense Disambiguation Challenges

Language is inherently extensible. Historical linguistics (the study of the change over time of linguistic systems) demonstrates that this can contribute to polysemy (words that have more than one meaning). Nouns can be employed as verbs: e.g. ‘n. skateboard’, ‘v. to skateboard’. Verbs can be employed as nouns: e.g. ‘v. to swim’, ‘n. go for a swim’. Either can usually be employed as adjectives or adverbs. There are generally small inventories of words / word classes that are not extensible in a language and they usually play specialized syntactic roles: e.g. determiners (a, the, those, etc). Other than that, very little other than community agreement limits the extension of language.

Linguistic extension continually adds nuance and novelty to language and means that language is always changing. Domain specific and socio-cultural specific employment of words adds and also deletes meaning from words. This means that the sense inventory associated with a word is malleable and often in a state of flex.

To highlight this the Merreiam-Webster online dictionary provides 20 glosses for ‘board’. The Cambridge Dictionary online however provides 35, with significant overlap. Dictionary.com provides 26 with significant overlap, and the Urban Dictionary provides 7 with no overlap. Even in sources (excluding Urban Dictionary) that have significant alignment between major senses, there is nuanced differences between definitions. For many researchers, this suggests that there may not ever be a crisp number of senses associated with a word and there is effectively no definitive means to create an authoritative sense inventory.

To highlight this Navigli observed differences in sensitivity to sense specification that leads to disagreement in manual disambiguation of corpora. He cites analyses that show Inter Annotator Agreement as low as 67- 80% (Navigli, 2009).

This has led some researchers to speculate that there is a theoretical ceiling to the effectiveness of automatic sense disambiguation systems (Popov, 2017).

* 1. Computational Word Sense Disambiguation Challenges

Some linguists estimate that comprehension of 95%-98% of words is necessary to understand text (Scmitt, Jiang, & Grabe, 2011). Currently, state of the art WSD systems achieve about 70% over all accuracy on general Word Sense Disambiguation tasks. (Raganato, Camacho-Collados, & Navigli, 2017).

When the research topic was first introduced in the 1940s, expert systems were employed to attempt WSD. These systems saw limited success as the field of linguistics itself was at that time still transforming from a somewhat prescriptive study of language to a descriptive discipline. In the late 20th Century, probabilistic models of language enjoyed significant increases in effectiveness but still did not come close to making generally effective in natural language processing. Since the beginning of this century deep learning techniques, notably variations of artificial neural networks predominate in increasing the effectiveness of WSD systems.

In addition to neural networks, fuzzy methods have also been employed. Due to the apparently fuzzy nature of word meanings themselves, fuzzy methods appear to lend themselves to WSD.

As with other areas of linguistics and semantics, incomplete understanding of the neural/linguistic processes that allow WSD in humans impedes progress on a computational solution.

In the papers discussed here, some rely on ‘knowledge based’ techniques. That is they employ some human curated reference that follows semantic guidance to determine senses similar spirit to expert systems. Others employ ‘supervised learning’, meaning they also employ human curated knowledge sources, but employ machine learning to conduct WSD. Still others employ ‘unsupervised learning. Meaning that they do not require human curated knowledge sources and but apply machine learning directly to data to discern WSD strategies.

The WSD problem is founded in the ambiguity of language. Not only can one ‘word’ have different meanings, it appears there is no way to discretely identify all of a words senses. In addition, the extensibility of language can lead to highly nuanced, even domain or social-group specific senses of words. On top of that within a single discourse it is entire possibly to encounter a fully novel employment of a word, and or have to entirely infer a words meaning from context (this is wholly the case with pronouns). These points and others frustrate computational efforts to conduct WSD.

1. Fuzzy Inference

In “Constructing Fuzzy Relations from WordNet for Sense Disambiguation” authors Christos Diou, Giorgos Katsikatos and Anastasios Delopolous (2006) attempt a ‘knowledge based’ solution to WSD. They’re method relies on Princeton University’s WordNet to generate sense candidates for a word and to select the best sense for a word in context.

* 1. WordNet

WordNet is a “lexical database of English Nouns, Verbs, Adjectives and Adverbs, grouped into sets of cognitive synonyms” (Princeton University, 2010). WordNet groups words into synonym sets (synsets) each expressing a distinct cognitive concept. (e.g. board.n.02 – a stout piece of sawn timber; made in a variety of sizes and used for many purposes, [board, plank]).

WordNet goes further to define semantic relationships between synsets. For nouns it maps hyperonymy/hyponymy (ISA) relations, holonymy/meronomy (PARTOF) relations. For verbs it maps hypernomy, troponymy (specification) and entailment relations. For adjectives it maps antynomy relations.

WordNet was first developed in the mid 1980s and the latest version 3.1 was released in 2011. WordNet has become an important lexical and semantic resource to both linguistics, computational linguistics and psycholinguistics (due to its application as an ontological database).

WordNet is hand curated, meaning, like dictionaries, lexicographers craft entries manually. It was originally based on the Brown Corpus but has since incorporated many other lexical databases.

The WordNet database encodes 155,327 words organized into 175,979 synsets and 207,016 word-sense pairs. Each synset is comprised of a list of ‘words’ or lemma (morphology is stripped from all lemma). The synset itself encodes semantic relationships (synonymy) and is accompanied by a gloss. Each synset is also accompanied by the lexical/semantic relationships described above. Wordnet also includes frequency data for senses gathered from source texts. When a word has multiple senses Wordnets attmempt to order them such that sense 1 represents the most frequent sense, sense 2 the next most frequent, etc.

WordNet is frequently employed in Part Of Speech tagging and Named Entity Recognition. However, the semantic knowledge it encodes also lends itself to informing WSD.

* 1. Generating Sets of Senses from WordNet

By design, each WordNet synset is a distinct cognitive entity and no two synsets will share the same meaning. The authors take each synset to represent a sense S. Each synset/sense S has a many to many relationship with words W. A word W may be associated with many synsets S­i (if it is polysemous). For example the word ‘board’ has has membership in 13 synsets (e.g. board.n.01 – ‘a committee having supervisory powers’, board.n.02 – ‘a stout piece of sawn timber’, etc.). Each synset S likewise may be associated with many words WS (e.g. board.n.02 [board, plank]).

The authors considered two senses Si, Sj to be related if there was a direct relationship in Wordnet between them Rt(Si, Sj). Where t is a type of relationship. For this work the authors considered the hyperonymy /hyponymy, holonymy/meronymy and domain relationships. The authors subjectively weighted each relationship based on their assessment of the importance of each relationship.

Hyperonymy = .4

Hyponymy = .9

Holonymy = .4

Meronymy = .9

Domain = .9

They used these weights to define a fuzzy set associated with sense Si that includes all senses Sj that have a direct relationship with Sj.

For example if Si = board.n.03 – ‘a flat piece of material designed for a special purpose’ has a hyponymy relationship with Sj = wallboard.n.01 – ‘a wide flat board used to cover walls or partitions’ then Rt=hyponymy(Si, Sj) = degree of membership dt = .9.

* 1. Generating Sets of Words for Senses

The authors did not consider relationships, Rt in WordNet to be transitive. They did however employ the transitive closure of a set, Rt+ to collect all the words w associated with a sense Si. The transitive closure of Si is the set of all senses that can be ‘reached’ via direct relationships. That is, if Si has a relationship with Sj, and Sj has a relationship with Sk, then Sk is in the transitive closure of Si.

Further, for each word w in the synonym set for a sense S, the authors assign a degree of membership that reflects how well w represents S. They employ WordNets frequency data to assign higher values to more frequent senses of a word. For example, ‘board’ is a member of 13 synsets, meaning it has 13 senses according to WordNet. These senses are ranked by frequency data. More frequent senses were assigned higher membership degrees. This, according to the author’s interpretation of WordNet, ‘board’ represents the synset board.n.01 – committee, better than it represents board.n.02. By the same frequency data, ‘plank’ represents the synset board.n.02 better than ‘board’ because board.n.02 is ‘plank’s number 1 sense (while it is ‘board’s number 2 sense).

In this manner, the authors can assign degrees of membership[[1]](#footnote-1) to all words associated with any synset. Using that, they expand the transitive closure Rt+ of a set Si to include all words associated with any sense in the transitive closure. This provides a fuzzy set WSi = Rt+(wij, Si) = dij for all relations t and words w that have a non-zero membership degree. That is WSi is a fuzzy set that represents all words associated with a sense Si.

* 1. WSD with WordNet

The authors use this framework to enable their WSD algorithm. The algorithm relies on two assumptions.

1. The sense of a word depends on the context in which it appears

The authors construct a context C by considering N words to the left or right of a target word w0. Syntax and punctuation are ignored. Consider the text: “Drywall is fragile, boards may crack if dropped”.

If we are trying to disambiguate ‘boards’ we must first remove morphology and reduce to ‘board’; we also strip all punctuation. If we set N = 3 we get a crisp context C = [drywall, is, fragile, w0, may crack, if]. Where wl=-N = ‘drywall’ and wl=N = ‘is’.

WordNet tells us that ‘board’ has 13 possible senses. These senses then comprise a fuzzy set of senses associated with target word w0. Each sense is further associated with a fuzzy set of words (as previously described) such that we have fuzzy sets of words WS1, WS2, … WSi… WS13.

The authors claim, based on assumption 1 that words in context C determine the correct sense of w0. They quantify this with their second assumption

1. The association between a word w in C and a sense S of the target word w0 can be quantified by the degree of membership of w in the fuzzy set of words associated with S.

So, if any word in the context C overlaps with a word in a fuzzy set WSi, that asserts that Si is a correct sense of w0. The authors define the extent of this assertion as equal to the membership of that word w in the fuzzy set WSi. That means that the membership value di of a sense Si in the fuzzy set of senses associated with a target word w0 = the membership value dij of a word in the fuzzy set of words associated with Si that is also found in context C.

‘board’ has 13 senses Si and 13 fuzzy sets WSi of words associated with each sense. The first sense S1 = board.n.01 – ‘committee’ has a fuzzy set of words WS1 associated with it, but none overlap with the words in C. The third sense S3 = board.n.03 – ‘flat piece of material’ has a fuzzy set of words WS3 that includes ‘drywall’. The degree of membership of ‘drywall’ in WS3 becomes the degree of membership of S3 = board.n.03 in Swo.

The authors use an iterative algorithm to calculate the membership of each sense Si in Swo. The sense with the greatest membership is selected as the correct sense.

* 1. Discussion

The authors ran their algorithm on a hand tagged corpus, SemCor 2.0. SemCor is a subset of the brown corpus and has been manually disambiguated against WordNet 3.0. The authors ran experiments varying fuzzy operators (t-norm/t-conorm), width of context C, depth of search from target word w0 and utilization of WordNet frequency data. Their best mean precision for polysemous words was 62.7% and 70.5% overall (including non-polysemous words). In some cases the authors’ work improves on N-gram models and comes close to meeting human Inter Annotator Agreement. However membership weights in there system are manually calculated and in part rely on subjective assessment. As well the disambiguation of senses relies only on the surface representation of a word within a context and does not consider the underlying senses of the surrounding context. Nonetheless the work is an inspired application of fuzzy inference in WSD.

1. Fuzzy C-Means Clustering

In “Word Sense Disambiguation in Hindi Language Using Hyperspace Analogue to Language and Fuzzy C-Means Clustering” Devendra, K Tayal, Leena Ahuja, and Shreya Chhabra apply unsupervised machine learning to WSD. Their technique relies on fuzzy clustering.

* 1. Crisp Clustering

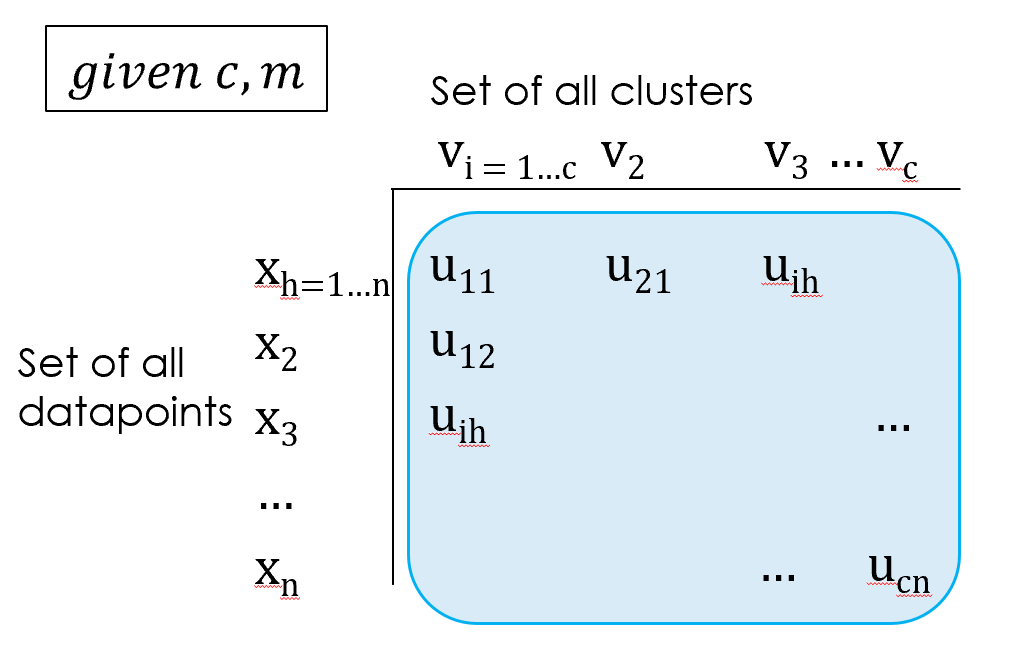
Crisp clustering, generally conducted with K-means clustering, segregates data into cells or clusters based on similarity to a idealized model. This is done by mapping data points into a vector space and using a Euclidian analysis of the vector space to associate each data point with the cluster it is closest to. This is done by calculating the Euclidian distance of a data point to a cluster center. K-means clustering conducts this iteratively, adjusting cluster membership and cluster centers until a desired threshold is met.

Importantly in crisp clustering every data point has a membership value of either 0 or 1 in a cluster and can belong to one and only one cluster.

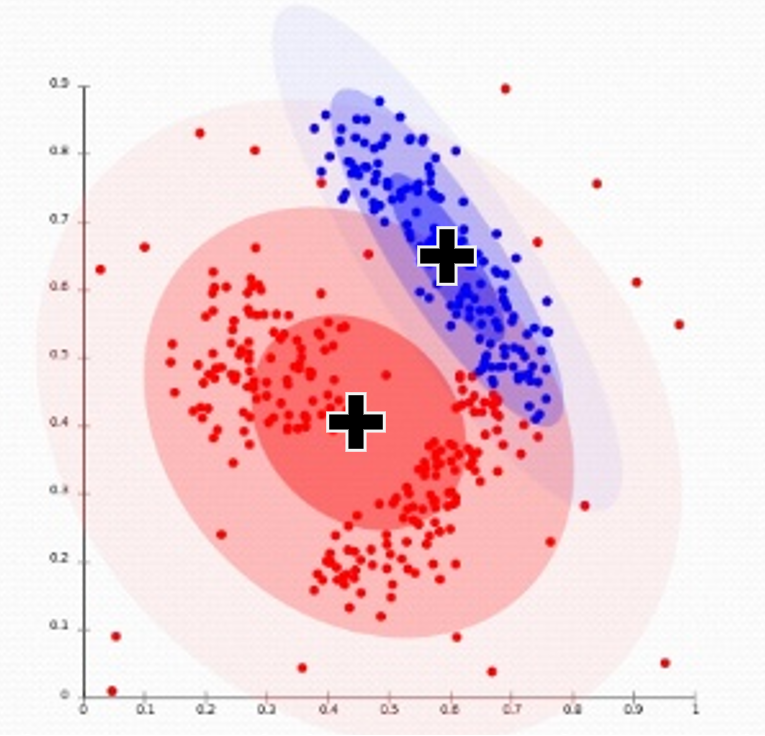
A close up of a map

Description automatically generated

* 1. Fuzzy Clustering

Fuzzy clustering, often carried out with Fuzzy C-Means clustering extends K-Means clustering by allowing data points to have fuzzy membership in more than one cluster. Datapoints can have a membership values of 0 to 1 in each cluster. In Fuzzy C-Means clustering, cluster centers are again based on the Euclidian center of a group of datapoints but now each datapoints influence on the center is weighted by its membership in the cluster.

d

Equation 1 provides the calculation for the centroid or center of a data cluster. This equation is derived from the Euclidian mean of all datapoints, and all every datapoints influence on the center is weighted by its membership in the cluster.

Every datapoint *x* from *h* = *1* to *n* is summed and weighted by its membership *uh* in cluster *vi*. Importantly, *x* is a vector not an x axis value. *uih* is raised to the power of *m*. *m* determines the ‘fuzziness’ of a set. If m = 1, clusters perform more like crisp clusters. As *m* increases, clusters become less crisp. *m* is often set to a value around 2.

Equation 2 is the objective function for Fuzzy C-Means clustering. This equation calculates the weighted mean of the distance of each datapoint *xh=1 to n* to each centroid *vi =1 to c*. This provides a measure J of the current analysis based on weights and centroids. Fuzzy C-Means clustering iteratively adjusts both datapoint weights and centroids to minimize J until a net change in the objective function reaches a desired threshold ε.

* 1. Hyperspace Analogue to Language

Fuzzy C-Means clustering relies on a vector encoding of every word. To achieve this the authors employ a Hyperspace Analogue to Language encoding. Like the WordNet solution, Hyperspace Analog to Language encoding relies fundamentally on an assumption that word senses are influenced by the the words in which they appear in context with.

In a vocabulary with *N* unique words, HAL begins with an *N* x *N* matrix. Stop words are removed to provide a *Q* x *Q* matrix. A window width *l* is chosen to examine 2*l* words on both sides of a target word. A weight W is developed for every word within the window of the target word

Thus every word is encoded as a vector of length Q that encodes all other words that it appears in context with. Every word vector can be considered a concept C in the source text that weights every word associated with it:

C =<Wcp1, Wcp2…Wcpq> (4)

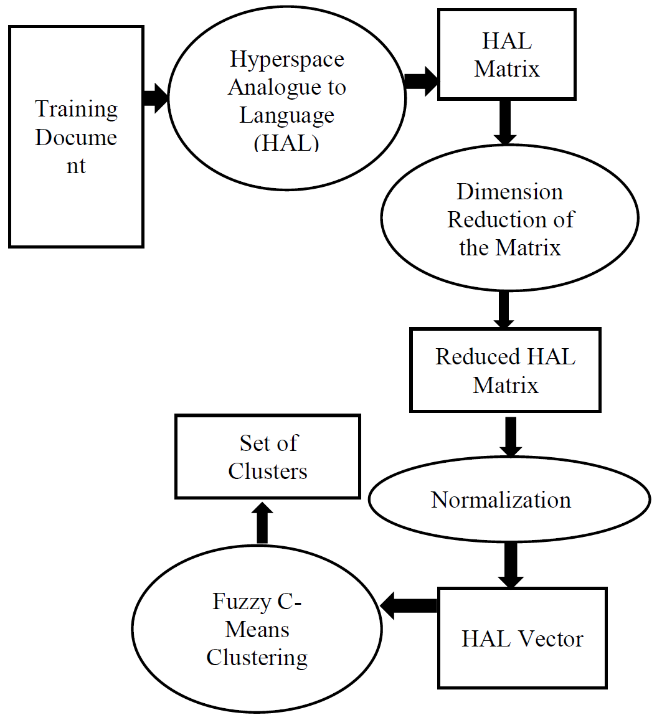
The authors normalize the weights in each vector to obtain the final HAL encoding of each word.

This encoding alone plots each word in a hyperspace where it will be spatially proximate to words it co-occurs with. Hyperspace Analogues to Language are important encoding tools in computational linguistics.

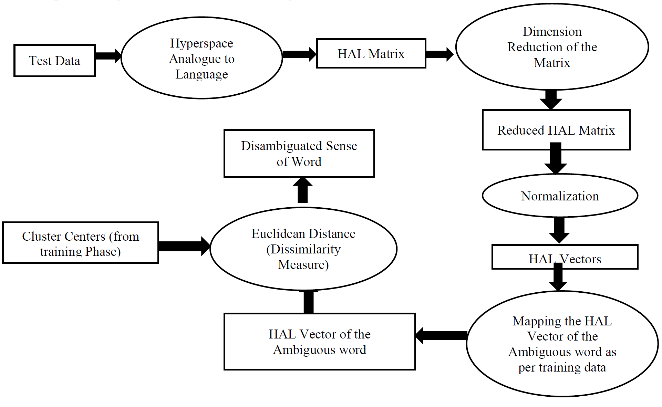
* 1. WSD with Fuzzy C-Means

The authors attempted to disambiguate the Hindi word *‘***तीर** which can mean either ‘arrow’ or ‘bank of a river’. The authors used both a training phase and a test phase to develop clusters and test their applicability in WSD.

* + 1. **Training Phase**

The authors used training data consisting of a text 3753 words long consisting of 91 unique words, 63 after stop words were removed. The data was encoded in a HAL matrix with Q = 63. The Hal vectors were provided to the Fuzzy C-Means algorithm for clustering into 2 sets with the assumption that the training data provides enough contexts of both senses of **तीर** to facilitate clustering around these two senses. Fuzzy C-Means clustering works for WSD specifically because a word vector can have membership in multiple clusters. The Fuzzy C-Means algorithm is used to generate two clusters.

* + 1. **Test Phase**

After clusters are generated in the training phase the authors took test data and manually created a Q dimensional HAL encoding for the target word based on its text context. They then measured the distance in hyperspace of this word vector to the two vectors developed in training. The closer centroid was selected as the representing the correct sense. The authors did not state exactly how they associated which sense with which cluster but it is assumed that they rely on the HAL encoding to inform the association.

* 1. Discussion

The authors report that they achieved a 79.16% success rate in disambiguating a corpus of Hindi Wikipedia articles and newspapers. It is not clear if they disambiguated all words or just **तीर.**

Either way, their results are significan and demonstrate the successful application of Fuzzy C-Means clustering in WSD.

A close up of text on a white background

Description automatically generated A shortcoming of this approach is that the number of senses to be disambiguated must be known and provided as a parameter to the Fuzzy C-Means algorithm. Potential difficulties with this were discussed in the opening introduction to WSD.

1. Kernel Fuzzy C-Means Clustering

In “Kernel Fuzzy C-Means Clusterting for Word Sense Disambiguation in Bio Medical Texts”, K. Ren and Y. F. Ren extend Fuzzy C-Means clustering into Kernel Fuzzy C-Means

* 1. Kernel Fuzzy Clustering

A picture containing food

Description automatically generatedKernel Fuzzy Clustering is a further extension of Fuzzy C-Means clustering. Sometimes the vector encoding of data points data points does not support the desired analysis. K and C-Means clustering can sometimes struggle with overlapping data because even in the fuzzy variant they rely on a Euclidian distance between a datapoint and the centroid of a cluster. When linear differentiability is not possible, K and C means clustering will struggle to develop the proper analysis.

Kernel Fuzzy C-Means clustering uses a transform Φ to map data into a higher dimensional space so that linear differentiability can be attained.

Many different transforms can be used for Φ but the most common are polynomial, gaussian (radial basis function) and hyperbolic tangent. The authors use the radial basis function in their work. If the transform Φ adheres to Mercer’s theorem then Kernel Fuzzy C-Means closely resembles Fuzzy C-Means clustering. The centroid function becomes:

And the objective function becomes:

(7)

The algorithm to employ these equations is essentially unchanged. *JΦ* remains a minimization target.

Because the Radial Basis Function satisfies Mercer’s Theorm (7) can be rewritten as:

(8)

Where K(x, y) =

* 1. Preprocessing

The authors only briefly discuss their encoding scheme. They are disambiguating words from the National Library of Medicine Word Sense Disambiguation (NLM-WSD) corpus. The authors state that this corpus contains 50 ambiguous biomedical terms in the corpus.

The authors extract 100 contexts for each ambiguous term from the NLM MEDLINE bibliographic database. The authors state that each context includes “title, id, abstract and the keyword’s position information’. It is not 100% clear exactly how all this information is converted into a feature vector. Although they do state that the ambiguous term in each context is hand tagged with sense information.

What the authors do discuss of their direct encoding states that they first remove stop words and low frequency words before calculating term frequency-inverse document frequency for each word (tf-idf)

tf-idf is a weighting scheme that attempts reflect a word’s relevance in a document. The basic idea is that term frequency, the number times a word appears in a document, the more it important it is and the greater influence it has on the document’s content. This however is moderated by the inverse document frequency, the number of documents a word appears in. This devalues stop words and linguistic function words like ‘a’, ‘the’, etc. The idea is that if a word appears in many or all documents, it is likely a function word with decreased semantic impact on a documents content.

* 1. Experimentation

The authors conducted several experiments. First they attempted to replicate experimental conditions with past work on word sense disambiguation in order to get objective comparison of their kernel fuzzy c means algorithm with previous WSD attempts.

During this they conducted both Fuzzy C-Means clustering and Kernel Fuzzy C-Means clustering in order to assess results of both methods. Finally they ran both algorithms on the full inventory of 50 ambiguous words

* 1. Sense Mapping

The authors did highlight the difficulties they had with mapping clusters to senses. In the previous work there were only two cluster senses and the appropriate sense could be determined from the HAL encoding.

It appears in this work that the authors did not use a HAL encoding. They extracted senses from the Unified Medical Language System, an authoritative and generally accurate resource of biomedical glosses. These senses provided the both the sense sense inventory for manual tagging as well as the number of clusters for clustering.

Clustering however only provides an analytical grouping in hyperspace. It does not map such groupings back to senses. Sense mapping in FCM and KFC clustering is a separate task. The authors used manual trial and error to arrive at the best sense to cluster mapping.

* 1. Discussion

In tests on limited data sets both FCM and KFC performed competitively with existing supervised learning systems. Generally KFC outperformed FCM and both generally outperformed existing methods.

When they tested their method against the whole 50 ambiguous words in the NLM-WSD, their KFC method achieved a success rate of 82% which is competitive with state of the art WSD systems.

1. Conclusion

The papers surveyed here provide proof of the applicability of fuzzy sets and fuzzy methods in word sense disambiguation. In general the unsupervised learning methods performed significantly better than the knowledge based system. Eventhough the methods were unsupervised though, system performance depended heavily on language encodings and source data provided.

Because of differences in encoding as well as testing against different data sets in different domains, it is difficult to objectively compare the performance of these systems. As stated both the FCM and KFC clustering appeared to significantly outperform the WordNet based system. As well, because the authors tested both in the last paper, it appears that KFC clustering does outperform FCM clustering.

This is likely not because the source encoding does not allow differentiability as it is already encoded in a high dimensional hyperspace. KFC likely further amplifies differentiability.

The availability of the application of fuzzy methods in WSD was somewhat sparse which was surprising. Fuzzy methods appear to lend themselves naturally to WSD and computational linguistics in general. Much of the current literature however is dominated by artificial neural networks.

Given the success of both, a fuzzy-neuro solution appears a common sense candidate for improving both approaches.

1. References

Diou, C., Katsikatos, G., & Delopolous, A. (2006). Constructing Fuzzy Relations fromWordNet forWord Sense Disambiguation. *2006 First International Workshop on Semantic Media Adaptation and Personalization (SMAP'06).*

Navigli, R. (2009). Word Sense Disambiguation: A Survey. *ACM Computing Surveys (CSUR)*.

Princeton University. (2010). *"About WordNet".* Retrieved from Princeton University: https://wordnet.princeton.edu/

Raganato, A., Camacho-Collados, J., & Navigli, R. (2017). Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1* (pp. 99-110). Valencia, Spain: Association for Computational Linguistics.

Ren, K., & Ren, Y. F. (2015). Kernel Fuzzy C-Means Clustering for Word Sense Disambiguation in BioMedical Texsts. *Journal of Digital Information Management*.

Tayal, D. K., Ahuja, L., & Chhabra, S. (2015). *Word Sense Disambiguation in Hindi Language Using Hyperspace Analogue to Language and Fuzzy C-Means Clustering.*

1. The authors used a sigmoid function followed by further normalization to provide final weights to each word<->sense relation. [↑](#footnote-ref-1)