SENTENCE SIMILARITY MEASURES TO SUPPORT WORKFLOW EXCEPTION HANDLING

A. Aldeeb, D. M. Pearce, K. Crockett and M. J. Stanton
Department of Computing and Mathmatics, Manchester Metropolitan University
Chester Street, Manchester, M1 5GD, U.K.

Keywords: Workflow Management System, Exception Handling, Case-based Reasoning, Sentence Similarity Measures.

Abstract: Exceptions occurrence in workflow systems is common. Searching in the past exceptions handlers' records, looking for any similar exception serves as good sources in designing the solution to resolve the exception at hand. In the literature, there are three approaches to retrieve similar workflow exception records from the knowledge base. These approaches are keyword-based approach, concept hierarchies approach and pattern matching retrieval system. However, in a workflow domain, exceptions are often described by workflow participants as a short text using natural language rather than a set of user-defined keywords. Therefore, the above mentioned approaches are not effective in retrieval of relevant information. The proposed approach considers the semantic similarity between the workflow exceptions rather than term-matching schemes, taking account of semantic information and word order information implied in the sentence. Our findings show that sentence similarity measures are capable of supporting the retrieval of relevant information in workflow exception handling knowledge. This paper presents a novel approach to apply sentence similarity measures within the case-based reasoning methodology in workflow exception handling. A data set, comprising of 76 sentence pairs representing instance level workflow exceptions are tested and the results show significant correlation between the automated similarity measures and the human domain expert intuition.

1 INTRODUCTION

A workflow management system (WFMS) is essentially a set of tools for modelling, enactment, and monitoring of business processes (Jablonski and Bussler, 1996). Workflow process definition (workflow schema) is the formal representation of a business process (Casati et al., 2000). The workflow schema is composed of activities (tasks) that collectively achieve the business goal. Workflow tasks are performed by workflow participants (Human or automated agent) according to their roles and the structure of the organization.

It is not guaranteed that designers always do a perfect job in defining a workflow type that totally represents all properties of the underlying business process (Hwang et al., 1999). In addition, the IT infrastructure of the WFMS and external factors can raise problems. Therefore, the occurrence of workflow exceptions is unavoidable and there is a need to handle those exceptions efficiently. Rule-based reasoning (RBR), Model-based reasoning (MBR) and case-based reasoning (CBR) are approaches being used to handle exceptions in workflow systems (Luo et al., 2003; Hwang et al., 2005). Workflow exceptions may require human intervention to establish proper handlers. Those handlers can be stored in a knowledge base to be used to handle similar exceptions in future in case of no available rules to handle them. Searching the exceptions handlers' records in the knowledge base, looking for any similar exception serves as good sources in designing the solution to resolve the exception at hand. (Luo et al., 2003; Montani, 2009; Hwang et al., 1999; Schmidt and Vorobieva, 2008; Grigori et al., 2001; Aldeeb et al., 2008). This can be achieved by applying CBR methodology to support the management of exceptions in business process execution. The main challenge in applying CBR to support exceptions handling in workflow systems is how to represent exceptions as cases, finding an effective retrieval mechanism of similar

Aldeeb A., Pearce D., Crockett K. and Stanton M. (2010).
SENTENCE SIMILARITY MEASURES TO SUPPORT WORKFLOW EXCEPTION HANDLING.
In *Proceedings of the 12th International Conference on Enterprise Information Systems - Artificial Intelligence and Decision Support Systems*, pages 256-263
Copyright © SciTePress

cases and the calculation of the similarity. Current applications of CBR in workflow exception handling use keyword-based retrieval system, concept hierarchies, and pattern matching and the use of decision trees to retrieve similar cases from the case database (Luo et al., 2003; Montani, 2009; Hwang et al., 1999; Schmidt and Vorobieva, 2008; Grigori et al., 2001). Some of these approaches depend on matching individual words in the current exception description with individual words in the textual material in the case base. On the other hand, the concept hierarchies approach is based on defining a concept of similarity and incorporates the notion of concept hierarchies. A concept hierarchy is a partial order of concepts, which indicates general-to-specific ordering where each case attribute has its own concept hierarchy. However, in a workflow domain, exceptions are often described by workflow participants as a short text using natural language rather than structural patterns of sentences. The workflow participants may express the same exception using quite different sentences in terms of structure and word content because of the diversity in human word usage. In addition, the same word can have different meanings. Therefore, irrelevant information may be retrieved and the relevant information may be missed. Some approaches restricting the allowable vocabulary, use intermediaries to generate indexing and search keys, or constructing explicit models of relevant domain knowledge. However, these approaches lack the flexibility to support the diversity in word usage of human, require expert users to generate indexing and search keys. These shortcomings and the limitations require an alternative approach which needs to consider the semantic similarity between the workflow exceptions rather than term-matching schemes. This motivates us to investigate the area of semantic sentence similarity measures and their potential application in workflow exception handling (Li et al., 2006; Feng et al., 2008; Aminul and Inkpen, 2008; Aliguliyev, 2009; Landauer et al., 1998).

This paper presents a novel approach to the application of sentence similarity measures within the CBR methodology to handle instance level workflow exceptions. A case study of the motor insurance process is used to prove the concept of our approach. The initial findings are encouraging and show that sentence similarity measures can be applied in the retrieval of relevant information in workflow exception handling in the knowledge base.

The rest of this paper is organized as follow: Section 2 discusses exception handling in workflow systems. Section 3 introduces some sentence similarity measures used in this research. In section 4, CBR as a methodology in the proposed approach is presented while section 5 illustrates a proof of concept prototype and case study. Finally, section 6 concludes and mentions some enhancements foreseen as a future work.

2 EXCEPTIONS HANDLING IN WFMS

WFMSs are designed to follow standard business processes and routine. However, these processes face the need to handle exceptions that fall outside the normal control flow (Casati et a., 2000). Exceptions occur commonly in workflows (Kumar and Wainer, 2005; Sadiq et al., 2005; Hwang and Lee, 2005). Workflow exception is any deviation from an ideal collaborative process that uses the available resources to achieve the task requirements in an optimal way (Klein and Dellarocas, 2000). There are four main causes of business process exceptions: system errors, data issues, external factors, and process design (Kelly, 2005). System errors can be independent of the transaction data and business logic and can be caused by underlying system problems, such as servers being down or services that are not available. Data issues can be missing, invalid or inconsistent data. External factors can trigger a process exception, such as that a specific item is out-of-stock or unavailable. Lastly, there can be process design issue that raise exceptions when specific cases need non-standard treatment for business reasons. Handling those exceptions depends on their type, severity and at what level they occur. Possible approaches to handle exceptions include ignore, retry, partial roll-back followed by forward execution, add some extra activities, delete some planned activities, or any change to the part of the workflow definition that not executed yet (Hwang et al., 1999).

Some workflow exceptions can be anticipated by the workflow designer, therefore they are called *expected exceptions*. However, others can not be anticipated and they are called *unexpected exceptions*. The expected exceptions are handled by rule-based reasoning. Those rules are characterised by the following components (Casati et a., 2000; Luo, et al., 2003):

- The *Event* part represents the symptoms of an exception

- The *Condition* is a boolean statement that checks if the symptoms is really an exception
- The *Action* describes the procedures that must be invoked to deal with the exception.

However, relying on predefined rules sometimes is not enough to deal with the unexpected workflow exceptions caused by ad-hoc changes. In this case, human intervention may be required to establish an appropriate handler. The successful exception handler can be stored for the future to deal with similar exceptions. Therefore, case-based reasoning can be applied to handle workflow exceptions by retrieving the similar exceptions handlers in the knowledge base. As we mentioned in the previous section, workflow exceptions are often described by workflow participants as a short text (sentence) using natural language rather than a set of user-defined keywords. This makes the process of building concept hierarchies and generating index keys of instance level workflow exceptions complicated. Table 1, shows an example of instance level workflow exceptions. Comparing those exceptions using sentence similarity measures directly is more practical, save time and effort that will be spend in building concept hierarchies and indexing keys. In the next section, sentence similarity measures are discussed.

3 SENTENCE SIMILARITY MEASURES

Sentence similarity measures have many applications, for example, Web page retrieval, text mining to discover unseen knowledge from textual database (Atkinson et al, 2004), text summarization (Erkan, and Radev, 2004), text categorization (Ko et al., 2004) and machine translation (Liu and Zong, 2004). Similarity computation techniques designed to detect the similarity between long texts are centred in shared words because similar long texts usually have a degree of co-occurring words. However, in short texts of sentence length, word co-occurrence may be rare or even null (Li et al., 2006) because people express similar meaning using quite different sentences.

The Latent Semantic Analysis (LSA) is one of the active researches in sentence similarity computation and information retrieval is (Landauer et al., 1998; http://lsa.colorado.edu/). LSA is based on statistical information of words in huge corpus. In LSA approach, a semantic space is automatically constructed for retrieval. The basic postulate is that there is an underlying latent semantic structure in word usage data that is partially hidden or obscured by the variability of word choice. A statistical approach is utilized to estimate this latent structure and uncover the latent meaning. Words, the text objects and, later, user queries are processed to extract this underlying meaning and the new, latent semantic structure domain is then used to represent and retrieve information. A set of representative words needs to be identified from a large number of contexts (each described by a corpus). A word by context matrix is formed based on the presence of words in context. The matrix is decomposed by singular value decomposition (SVD) into the product of three other matrixes, including the diagonal matrix of singular values. The diagonal singular matrix is truncated by deleting small singular values. In this way, the dimensionality is reduced. The original word by context matrix is then reconstructed from the reduced dimensional space. Through the process of decomposition and reconstruction, LSA acquires word knowledge that spreads in context. When LSA is used to compute sentence similarity, a vector for each sentence is formed in the reduced dimension space; similarity is then measured by computing the similarity of these two vectors. Because of the computational limit of SVD, the dimension size of the word by context matrix is limited to several hundred. As the input sentences may be from an unconstrained domain (and thus not represented in the contexts), some important words from the input sentences may not be included in LSA dimension space

Li et al. (2006) proposed a method named (STASIS) that can be used generally in applications requiring sentence similarity computation. This method is fully automatic and adaptable across a range of potential application domains. The proposed method dynamically forms a joint word set only using all the distinct words in the pair of sentences. Then, for each sentence:

- A raw semantic vector is derived with the using a lexical database.
- A word order vector is formed for each sentence using information from lexical database.
- The significance of the words and their contribution to the sentence meaning is weighted using information derived from corpus.
- By combining the raw semantic vector with information content from the corpus, a semantic vector is created for each of the two sentences.
- Semantic similarity is computed based on the two semantic vectors.
- An order similarity is calculated using the two order vectors.

- Finally, the sentence similarity is derived by combining semantic similarity and order similarity.

In the evaluation phase of the STASIS, a set of sentence pairs are collected from a variety of articles and books in computational linguistics. An initial experiment on this data illustrates that the proposed method provides similarity measures that are fairly consistent with human knowledge (Li et al., 2006).

Both LSA and STASIS measures are used in implementation of the proposed approach. The next section illustrates the case-based reasoning methodology in handling workflow exceptions and applying sentence similarity measures in the retrieval phase.

4 CASE-BASED REASONING METHODOLOGY

CBR is a reasoning paradigm that exploits the specific knowledge of previously experienced situations, called *cases*, to learn and generate hypotheses about new situations (Montani, 2009; Shiu and Pal, 2004a). The use of CBR can reduce the amount of effort needed to formalize the knowledge, since representing a real world situation as a case is often simple. The use of CBR facilitates an automatic acquisition and increases the operative knowledge, without requiring a hard and time consuming formalization of knowledge itself, as it required by other methodologies, such as rule-based or model-based reasoning (Montani, 2009).
However, rule-based and model-based reasoning are more effective for applications where theory, not experience, is the primary guide to problem solving and the solutions are designed for a specific problem and are difficult to be adapted (Limam et al., 2003). A case consists of problem description and case solution. CBR can therefore be described by the CBR-cycle which comprises four activities (Watson, 1999):

1- Retrieve similar cases to the problem description
2- Reuse a solution suggested by a similar case
3- Revise or adapt that solution to better fit the new problem if required
4- Retain the new solution once it has been confirmed or validated

For complicated real world applications there are some degree of fuzziness and uncertainty that almost always encountered (Shiu and Pal, 2004b). AI techniques such as fuzzy logic, neural networks and genetic algorithms are helpful in areas where uncertainty, learning and knowledge inference are part of the system application (Shiu and Pal, 2004a; Pal et al., 2004; Jeng and Liang, 1995; Pal and Shiu, 2004). In this research we suggest the sentence similarity measures be part of the CBR to support cases similarities calculation and retrieval phase of the CBR-cycle.

Our approach is to maintain records about the past experience of handling exceptions. Those records form cases in the knowledge base to be used to handle exceptions which need to be managed in similar way, but may occur in different workflow instances. The structures of the cases that represent exceptions in the knowledge base are described in the next section.

4.1 Exception Representation as a Case

The structure of the workflow exception cases is inspired by and adapted from the work of Hwang et al. (1999). Each workflow exception case consists of three main components; *Exception Information Block (EIB), Exception Handler Block (EHB), and Workflow Instance Block (WIB).*

The EIB represents the problem description part of the case and contains the following main exception features or attributes:

- Exception Description: Is the semantic information that describes the exception by the workflow user. This description takes the form of short text (sentence length).
- Status: Is the status of the workflow instance that raises the exception e. g., initiated, active, suspended, terminated or complete.
- Workflow Participant: is the performer of the activity that raises the exception. This can be automated agent or human.
- Workflow version: represents to which version of the workflow schema the affected workflow instance belongs.
- Time: indicate when the exception occurs
- Frequency: is a number indicates how many times this exception case is successfully used. When this number reaches a certain threshold value, it will trigger the evolution of the process model.

The EHB represents the solution part of the case and indicates the action to be taken to handle the exception. This can be an automated action or manual action requiring user intervention. Generally, the action can be under one of the following categories:

- Maintaining the workflow normal behaviour e. g.: Ignore record, notify or add resource.
- Modifying the workflow behaviour e. g.: retry, suspend, modify, remove, change sequence, terminate, re-assign and delay.
- Modify the workflow schema e. g. making a new version and the affected workflow instance will follow the new version.

The third block, WIB, contains the workflow instance itself which consists of the data and control attributes of the workflow. The control attributes are used by the exception handler.

4.2 The Exception Handling Mechanism

The exception handling procedure is as follows:

- The workflow participant propagates the exception raising case to the Exception Handling Coordinator (EHC) in the WFMS accompanied by brief event description (short text of sentence length, 15 to 25 words). This sentence is the semantic information that describes the type of the exception.
- Upon receiving the exception case, the EHC creates a temporary case template (TCT). The TCT contains two blocks; workflow instance information block (WIB) and exception information block (EIB).
- The event description in the TCT will be compared with the exception description attributes in the case database using sentence similarity measures. Similar cases will be retrieved based on a certain similarity rang established by the human domain expert.
- The most similar case with the highest overall matching mark (based on an established threshold value defined by the domain expert) is chosen and its solution will be applied with or without adaptation.
- If no adaptation the case usage frequency field is incremented and the TCT will be deleted.
- In case of adaptation, a new case will be created by adding the modified exception handler block (EHB) to the TCT and storing it as a new case in the case base
- If there is no similar case found, the TCT will be stored as a new case in the case base and EHB will be added. The new EHB is based on the judgement of the process engineer and the domain expert
- Case database maintenance will be performed regularly merging highly similar cases or removing unused cases.

As we mentioned above, the retrieval of similar cases depends on the sentence similarity measure between the current exception description and the exceptions descriptions in the case database. In addition, some CBR approaches e. g. (Weber and Wild, 2005) use conversational scheme (question/answer) with workflow participant to find the best match between the case at hand and a number of similar cases in case database. As these questions and answers are short texts (sentence length), again sentence similarity measures will be very useful in retrieval of the best match. In next section, a prototyping of the system and a case study are presented.

5 A CASE STUDY

Car insurance claims handling is a process which needs to be automated and managed efficiently and be adaptable to the changing circumstances. However, a fully automated solution for claim processing cannot handle exception claims and does not have real-time situational awareness capabilities. Therefore, an effective exception handling mechanism is needed. This is achieved by a dedicated exception handling coordinator which can be part of the workflow server or a separate node connected with the workflow administrator.

To better illustrate how the proposed exception handling mechanism works, a prototype of workflow management system is developed to automate a motor insurance claim process. The process consists of 11 tasks involving 4 workflow participants representing four roles in the insurance company. These roles are: claim team, legal team, finance team and technical team. Figure 1 shows the motor insurance claim process at the modelling phase in Petri notation. The workflow management system is implemented in Java, and relies on Microsoft Access database.

To prove the concept presented in this paper the WFMS prototype carries out its run-time function including the exception handling routine which includes:

a) Instantiating a number of workflow instances (claims) and coordination of tasks between the workflow participants is practised
b) Exceptions are generated by the workflow participants in the running instances and exception handling procedures mentioned early in the paper are examined.

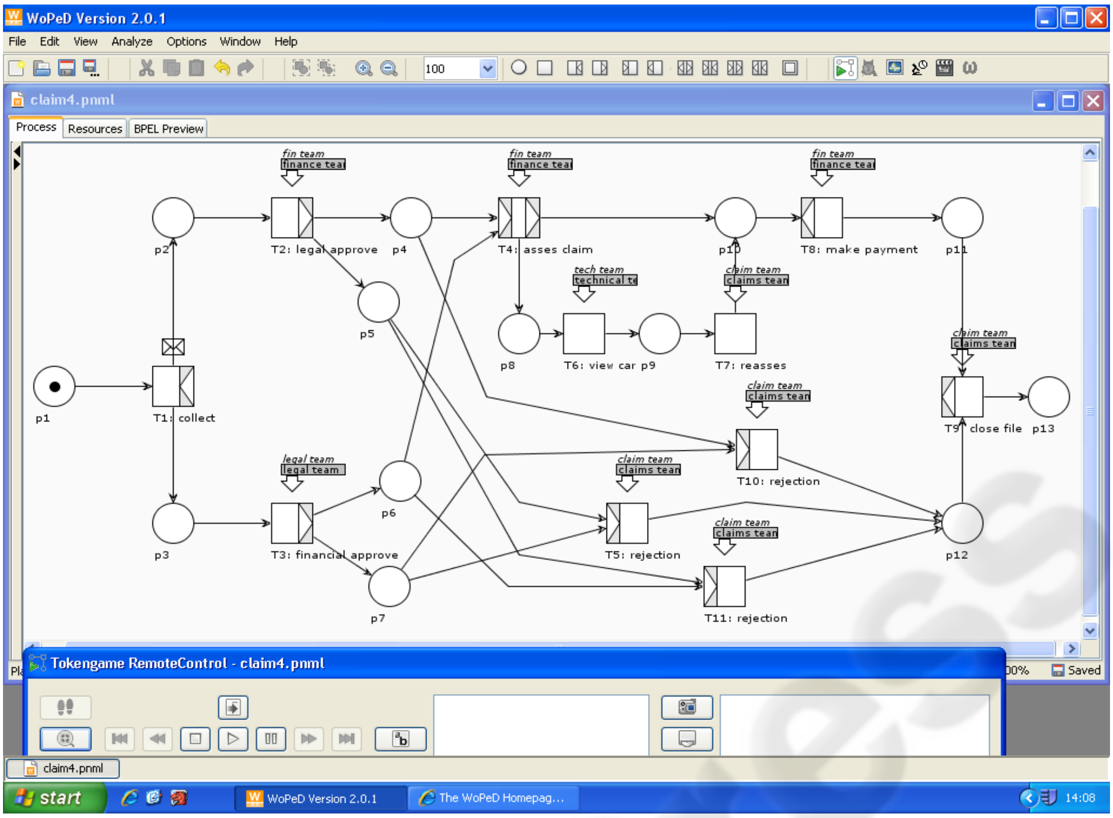

Figure 1: Insurance claim process in Petri net presentation.

Table 1: Sample of some exception pairs and their similarity measures.

Exception Pair	Similarity		
	Human D. E.	LSA	STASIS
The insured car is being used for a purpose other than that shown in the description of use in the Certificate. The damage happened while the car is being used for a purpose not mentioned in the certificate.	0.90	0.83	0.71
Insured car has been modified for performance. The car has been modified from the manufacturer's standard specification.	0.80	0.86	0.75
Insured car is being used for racing. The car is being used for rally.	0.90	0.95	0.81
Inaccurate claim expenses provided by the loss adjuster. Additional damage has been found by the loss adjuster.	0.60	0.52	0.62

The generated exceptions in the testing are real insurance claims exceptions collected and adapted from different insurance companies websites. 92 exceptions pairs are tested to compute the similarity between them. These exceptions are passed to the STASIS and LSA programs to compute the similarity. The results obtained from the above algorithms are compared to the similarity judgement of the human domain expert. Table 1 shows an example of the exception pairs and the similarity measures.

Figure 2 shows the similarity measurement between samples of 76 exceptions pairs using LSA while Figure 3 shows the similarity measured by a human domain expert. Similarity falls within a range of zero to one (where zero is totally dissimilar and one is an exact match). As we can see from the figures, the results show significant correlation between the automated similarity measures and the human domain expert intuition. Around 60% of the exceptions pairs' measures are very close or close to the human judgement. The average response time to compute the similarity between each pair of exception sentence is 0.065 second.

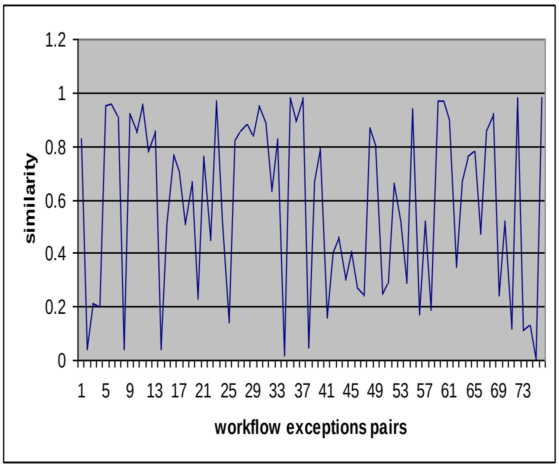

Figure 2: Exceptions pairs' similarity measurement using LSA.

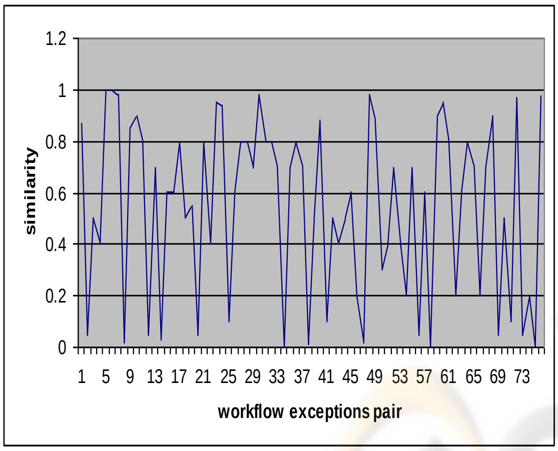

Figure 3: Exceptions pairs' similarity measured by human domain expert

6 CONCLUSIONS

Workflow exceptions are often described by workflow participants as a short text (sentence length) in natural language. Therefore, sentence similarity measures can be used to compute the similarity between the exception at hand with the exceptions stored in case database in order to find the proper exception handler. This paper presented a framework to apply semantic sentence similarity measures within case-base reasoning paradigm in workflow exception handling. We believe that sentence similarity measures are capable techniques in helping retrieve appropriate cases. This research provides the starting point in the semantic sentence similarity application in WFMSs. As the sentence similarity measurement techniques are still active research area, we expect new enhancements to those techniques which will provide more accurate results.

We concentrated in this paper on presenting the ideas, the concepts, the architecture and the initial experimental results of our approach. Further work and publication will include the following:

- Statistical and mathematical modelling of the proposed approach and the obtained results to compute the accuracy and the optimization of our measurements
- Study the effect of the noise (noisy sentence which includes missed words, grammatical error and spelling error) and how the similarity measurement techniques cope with this
- Apply the proposed approach to different business processes from different domains.

REFERENCES

Aldeeb, A., Crockett, K. and Stanton, M., 2008. An Inter-organizational *Peer-to-Peer Workflow Management System. In Proceeding of the 10th International Conference on Enterprise Information Systems (ICEIS-2008)*, pp. 85-92, Barcelona, Spain

Aliguliyev, R., 2009. A New Sentence Similarity Measure and Sentence based Extractive Technique for Automatic Text Summarization. *Expert Systems with Applications*, 36, pp. 7764-7772.

Aminul, I. and Inkpen, D., 2008. Semantic Text Similarity Using Corpus-based World Similarity and String Similarity. *ACM Transactions on Knowledge Discovery from Data*, Vol 2, No. 2, Article 10.

Atkinson, J., Mellish, C. and Aitken, S., 2004. Combining Information Extraction with Genetic Algorithms for Text Mining. *IEEE Intelligent Systems*, Vol. 19, no. 3,

Casati, F., Ceri, S., Paraboschi, S. and Pozzi, G., 2000. Specification and Implementation of Exceptions in Workflow Management Systems. *ACM Transactions on Database Systems*, Vol. 24, No. 3, pp. 405-451

Erkan, G. and Radev, D., 2004. LexRank: Graph-Based Lexical Centrality As Salience in Text Summarization. *J. Artificial Intelligence Research*, vol. 22, pp. 457-479

Feng, J., Zhou, Y. and Trevor, M., 2008. Sentence Similarity based on Relevance. *Proceedings of IPMU'08*, pp. 832-839

Grigori, D. et al., 2001. Improving Business Process Quality through Exception Understanding, Prediction, and Prevention. *Proceedings of the 27th VLDB Conference*, Roma, Italy.

Hwang, G. and Lee, Y., 2005. A Flexible Failure-Recovery Model For Workflow Management Systems. *International Journal of Cooperative Information Systems*. Vol. 14, No. 1 pp. 1-24

Hwang, S., Ho, S. and Tang, J., 1999. Mining Exception Instances to Facilitate Workflow Exception Handling. *Proceedings, 6th International Conference on Database Systems for Advanced Applications*, pp. 45-52.

Jablonski, S. and Bussler, C., 1996. *Workflow Management: Modelling Concepts, Architecture and Implementation.* International Thomson Publishing.

Jeng, B. and Liang, T. (1995) Fuzzy Indexing and retrieval in case-based Reasoning Systems. *Expert Systems with Applications*, 8, no. 1, pp. 135-142

Kelly, D., 2005. *Where Do Business Process Exceptions Come From?* EbizQ, http://www.ebizq.net/topics/bpm/features/6382.html?&pp1, viewed on 10/06/2009 12:16

Klein, M. and Dellarocas, C., 2000. A Knowledge-based Approach to Handling Exceptions in Workflow Systems. *Computer Supported Cooperative Work,* 9: pp. 399-412

Ko, Y. Park, J. and Seo, J., 2004. Improving Text Categorization Using the Importance of Sentences. *Information Processing and Management,* Vol. 40, pp. 65-79

Kumar, A. and Wainer, J., 2005. Meta Workflow as a Control and Coordination mechanism for Exception Handling in Workflow Systems. *Decision Support Systems* 40, pp. *89-105.*

Landauer, T., Foltz, P. and Laham, D., 1998. Introduction to Latent Semantic Analysis. *Discourse Processes,* vol. 25, nos. 2-3, pp. 259-284

Li, Y. et al., 2006. Sentence Similarity Based on Semantic Nets and Corpus Statistics. *IEEE Transactions on Knowledge and Data Engineering, 18,* pp. 1138-1150.

Limam, S., Marir, E. and Reijers, H., 2003. *Electronic Journal on Knowledge Management,* Vol. 1, 2, pp. 113-124

Liu, Y. and Zong, C., 2004. Example-Based Chinese-English MT. *Proceedings of 2004 IEEE Conference, Systems, Man, and Cybernetics,* vols. 1-7, pp. 6093-6096

Luo Z. et al., 2003. Exception Handling for Conflict Resolution in Cross-Organizational Workflows. *Distributed and Parallel Databases*, 13, 271-306.

Montani, S., 2009. Prototype-based Management of Business Process Exception Cases. *Applied Intelligence,(online version),* Springer Netherlands

O'Shea J., Bandar, Z., Crockett, K. and McLean, D., 2008. A Comparative Study of Two Short Text Semantic Similarity Measures. *LNAI 4953*, pp. 172-181.

O'Shea, K., Bandar, Z. and Crockett, K., 2008. A Novel Approach for Constructing Conversational Agents using Sentence Similarity Measures. *Proceedings of World Congress on Engineering, International Conference on Data Mining and Knowledge Engineering, London,* pp. 321-326.

Pal, S. and Shiu, 2004. *Foundations of Soft Case-based Reasoning*. John Wiley, NY

Pal, S., Polkowski, L. and Skowron, A., 2004. Rough-Neural Computing: Techniques for Computing with Words. Springer Verlag, Berlin

Sadiq, S. Orlowska, M. and Sadiq, W., 2005. Specification and Validation of Process Constraints for Flexible Workflows. *Information Systems*, 30, pp. 349-378.

Schmidt, R. and Vorobieva, O., 2008. Using Case-based Reasoning To Explain Exceptional Cases. Proceedings of 10th International Conference on Enterprise Information Systems, pp. 119-124.

Shiu, S. and Pal, S., 2004. Case-Based Reasoning: Concepts, Features and Soft Computing. *Applied Intelligence 21*, pp. 233-238

Shiu, S. and Pal, S., 2004. Case-based Reasoning: Concepts, Features and Soft Computing. *Applied Intelligence21*, pp. 233-238

Watson, I., 1997. *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. Morgan Kaufmann Publishers, San Francisco, USA.

Weber, B. and Wild, W., 2005. Conversational Case-based Reasoning Support for Business Process Management. *American Association for Artificial Intelligence (*www.aaai.org*).*