Achieving Meaning Understanding in E-Marketplace through Document Sense Disambiguation

Jingzhi Guo, Guangyi Xiao

Achieving Meaning Understanding in E-Marketplace through Document Sense Disambiguation

Jingzhi Guo and Guangyi Xiao

Department of Computer and Information Science, University of Macau,
Av. Padre Tomás, Pereira, S.J., Taipa, Macau
{jzguo, ya97409}@umac.mo

Abstract. E-marketplace has a very important requirement of achieving mutual meaning understanding between sellers and buyers. To meet this requirement, this paper has proposed a novel SD-DSD approach, which enables to disambiguate senses between a sender and a receiver for their sent and received documents. This approach has developed five novel strategies for sense disambiguation. Based on them, new document representation models for message exchange are devised, together with their sense consistency control procedures and sense interpretation evaluation method.

Keywords: Meaning understanding, sense disambiguation, business document, e-marketplace, XML Product Map (XPM).

1 Introduction

E-marketplace is a rapidly evolving research area and has been received significant attentions from academia and industry (e.g., [3, 10, 16, 13], emarketservices.com). It is a common business information space (CBIS) and is an infrastructure of e-market where buyers and sellers at various enterprise information systems conduct business online [4]. It has a very important requirement of achieving mutual meaning understanding between sellers and buyers. *Meaning understanding* refers to reaching a certain level of semantic agreement between the communicating parties of human through underlying disparate messaging systems over various networks or Internet. This requirement for meaning understanding has to be implemented in e-marketplace. This is because sellers and buyers can conduct business if and only if they well understand what they are talking about and have no misunderstanding on their exchanged messages with each other.

Consider a case as follows: Seller S sends a valid offer of fridge to Buyer B and Buyer B confirms the offer by sending back an offer acceptance. In this legally valid offer-acceptance business cycle, the Seller's offer is made in Table 1 and the Buyer's offer acceptance is made in Table 2. Both tables are generated based on their local databases and the messages in exchange. Now, the problem happens such that Seller S deems that it sells a mini household refrigerator in US$250, but Buyer B believes it confirms an offer of camping fridge only worth of HK$250. Definitely, this is a legally-flawed offer-acceptance cycle and will cause legal consequences.

Technically speaking, the above case can be easily avoided if the offer-acceptance cycle is processed by human. Nevertheless, when a trading process is automatically handled by autonomously developed software systems, the business document sense disambiguation becomes a tough research problem and must be resolved.

Table 1: A valid offer of Seller S

Offer No.	Commodity	description	Quantity	Price	Total
S111	fridge	orange, low temperature	100 pieces	$250	$25,000
This offer is valid before 2009/12/15.					

Table 2: A valid offer acceptance from Buyer B

Acceptance No.	Commodity	description	Quantity	Price	Total
B222	fridge	orange, low temperature	100 pieces	$250	$25,000
This acceptance confirms the offer no. S111 on 2009/12/10.					

Trading process automation is a very important topic and long been recognized in e-commerce research area [8, 12, 14, 15]. It is the design foundation of modern e-marketplace that saves business costs and increases efficiency of sellers and buyers. At the core of trading process automation is the document sense disambiguation that helps reach mutual meaning understanding between buyers and sellers for their smooth conducting electronic business. *Document sense disambiguation* (DSD) is a specialized research of word sense disambiguation (WSD) [11], focusing on identifying the meaning of a document to generate a correctly interpreted document. For example, in the above case, in order to avoid a legally-flawed offer-acceptance cycle, Buyer B must equip with a DSD tool to test whether its interpretation on the document of Seller S is accurate, following the original meaning of Seller S. Document sense ambiguities during interpretation are often caused by document systems autonomy, which can happen in the levels of document design, document communication and document execution [5:15-18]. Thus, the sense disambiguation methods can also be classified in these three levels. In addition, the target document for sense disambiguation can be an unstructured document (UD) (e.g. an article in plain text) or a structured document (SD) (e.g. an invoice or an offer sheet in tabular form). Traditionally in natural language processing, a most important task is to enable machines to process unstructured textual information and transforms them into data structure that can be analyzed to determine the underlying meaning. Such computational identification of meaning for words in context is called *word sense disambiguation* (WSD), which is an AI-complete problem [11] when a text is an unstructured document (UD).

This paper aims to propose a novel SD-based document sense disambiguation (SD-DSD) approach to resolve the ambiguous interpretation problem that causes the gap of meaning understanding between sellers and buyers. It resolves the problem in design phase and attempts to turn an AI-complete problem of WSD [11] into a relatively easy problem through constructing business documents in a uniquely identified concept hierarchy. It assumes that both sellers and buyers in e-marketplace are sufficient to utilize a certain structured document (SD) for achieving mutual meaning understanding during trading communication. It is, thus, contented that a solution to SD-based document sense disambiguation suffices to achieve the goal of

this paper if and only if an SD-based document can be transformed into a uniquely identified concept hierarchy, initially described in [5, 6] for constructing interoperable electronic product catalogues.

The rest of the paper is arranged as follows. Section 2 proposes a novel approach of SD-based document sense disambiguation to achieve mutual meaning understanding between sellers and buyers in e-marketplace. Section 3 implements the proposed MD document data model in XPM format. In Section 4, an evaluation method is suggested to evaluate the interpretation accuracy for any incoming MD document. Section 5 briefly describes the related work of the proposed approach. Finally, a conclusion is made together with contributions and future work.

2 Structured Document Based Sense disambiguation

The traditional sense disambiguation often targets at unstructured free text. It takes a whole piece of free text as the input and then tries to make sense on what the input text means. The hardness of the problem is AI-complete [11]. However, under the circumstances of e-marketplace trading activities, a free text is not necessary for conveying the business meaning between sellers and buyers. A structured tabular document suffices for meaning understanding in most trading cases. For example, an inquiry can be designed as a structured tabular inquiry sheet, constituting a set of concept pairs of an abstract concept and its reified concept such as (color, red) or (price, 100). This provides an opportunity to turn an AI-complete problem of WSD into an easier problem that is suitable to be solved to identify explicit and consistent meanings of a structured document. We shall call such an easier problem as a *one-to-one match problem*, which means that as long as any concept of a text matches a one-to-one correspondence in an interpreted text, the text is capable of sense disambiguation. Thus, the problem of WSD is transformed into a problem of SD-based document sense disambiguation (DSD) with the task of finding a one-to-one match-able document, in which each concept is explicit and consistent.

2.1 Problem Description

By observation, most e-marketplace documents are in tabular form that is structured. A one-to-one concept match problem, which could describe the senses ambiguity of a *structured tabular document* (SD), can be described as follows:

$$\text{Structured tabular document (SD)} \Rightarrow \text{One-to-one match document (MD)} \qquad (1.1)$$

This problem states that, if a concept match document MD could be found to have a one-to-one correspondence with SD document in concept match, then the one-to-one concept match problem could be solved as (1.1). Given the problem (1.1), our research task is to find a one-to-one match document MD from an existing structured tabular document SD. For example, given a table SD1 as shown in Table 1, we need to find a one-to-one match document MD1 as shown in Table 3, where each concept of MD1 can exactly match with each concept of SD1.

Table 3: An ideal MD1 file for SD1 as shown in Table 1

Concept	Word	Definition
1	offer	A scheme of payment for providing products
2	No.	A series of numerals or symbols used for reference or identification
3	S111	A particular series of numerals or symbols used for reference or identification
4	commodity	Something useful that can be turned to commercial or other advantage
5	fridge	An appliance for storing food or other substances at a low temperature.
6	description	The act, process, or technique of describing
7	orange	A kind of color
8	low	Below average in degree, intensity, or amount
9	temperature	A specific degree of hotness or coldness to a standard scale
10	quantity	A specified number
11	1000	A numerical value
12	pieces	A unit of quantity
13	price	The amount as of money, asked for or given in exchange for something else
14	$	United States dollar
11	250	A numerical value
16	total	A sum of all parts
14	$	United States dollar
11	25,000	A numerical value

The MD1 of Table 3 suggests a mapping A that in document SD1 only one sense S can be assigned to each word $w_i \in$ SD1 such that:

$$| A(i)_S | = 1 \ (i \in 1 \ldots n). \tag{2.1}$$

where i refers to a word w_i and s refers to the corresponding senses of i. The $A(i)_S$ means that every w_i in an SD1 corresponds to its senses to derive a map A. □

If this $| A(i)_S |= 1$ holds, we say that the sense of SD1 is fully disambiguated or the interpretation of SD1 could be 100% accurate. Generically, achieving $| A(i)_S |= 1$ for SD will, thus, solve the problem of SD-based document sense disambiguation.

2.2 Strategies of Achieving | A(i)S | = 1

To achieve (2.1) of $| A(i)_S | = 1$, this paper proposes five strategies, which are concept identification, concept hierarchy, concept atomization, concept pairing, and concept matching. In the following, we discuss these strategies one by one to show how we disambiguate the sense of a tabular SD document to 100% accuracy.

Strategy 1(*Achieving sense uniqueness by concept identification*): Given any word $w \in$ SD and any concept $c \in$ MD, w is uniquely sensed if and only if $w \Leftarrow c$ and $c \Leftarrow rt$, where rt is a unique identifier. □

By this strategy, the sense of each word has been assigned a corresponding uniquely identified concept. This makes it possible to construct a one-to-one match document, where word sense can be disambiguated. For example, the concept like "orange" in MD1 is uniquely identified by "7" and it will not be interpreted as fruit.

Nevertheless, unique concept identification is not enough. In real practice, many concepts are group concepts such as "price", which means "price(currency, amount, unit)". In fact, the flawed offer-acceptance cycle of the introduction case, shown in Table 1 and Table 2, happens due to the fact that "price" concept has not been well defined by missing the clear definition of "$". In many existing databases, the use of

"$" or omission of "$" is popular since the database designers assume a local context in data design. Such implicit concepts have to be explicitized when accurate sense of word is required. Strategy 2 is a method of resolving this problem.

Strategy 2 (*Achieving sense explicitization by concept hierarchy*): Given any word $w \in$ SD and a set of concepts c, c_1^1, c_i^2, …, c_i^k, …, $c_i^n \in$ MD, w is explicitly sensed if and only if $w \Leftarrow c$ and $c = (c_1^1, c_i^2, \ldots, c_i^k, \ldots, c_i^n)$, where c is a hierarchy with c_1^1 as the root node and c_i^k as a descendant node (i, k refer to a sibling and a level of hierarchy, respectively). □

Strategy 2 assumes that every concept is a group concept by default, though most concepts only contain only a root node concept such that $c = c_1^1$. With this assumption, it requires every concept to be explicitized by decomposing itself into member concepts until any member concept contains only a root concept. This strategy can effectively eliminate implicit concepts that cause interpretation ambiguity, for example, "price = price(currency, amount, unit)".

Strategy 3 (*Achieving sense independence by concept atomization*): Given any two words w, $w' \in$ SD and any two concepts c, $c' \in$ MD, w is independently sensed if and only if $w \Leftarrow c \parallel w' \Leftarrow c'$, where "||" is an independence relationship. □

This strategy is to ease the execution complexity of any concept. A uniquely, explicitly and independently sensed word remains the same sense in whatever place even outside of the document. The principle of concept atomization states "What you are saying is what you have said". For example, an independently sensed "orange" in MD1 can be permanently recognized as "fc: orange, cid: boo.com::docDB#S11.7, rt:voc.com::vocDB#2222". The sense of the word can always be disambiguated against the reference vocabulary in voc.com and found in boo.com as historical archive unless they have been deleted. By this strategy, everything becomes a history. Apparently, it well fits for business needs of tracking and verification from the perspective of legality.

Strategy 4 (*Achieving sense reification by concept pairing*): Given any two words w_1, $w_2 \in$ SD and any two concepts c_1, $c_2 \in$ MD, w_2 is a reified sense of w_1 if and only if $\exists 1$ *Pair*($w_1 \Leftarrow c_1$, $w_2 \Leftarrow c_2$) and $c_1 \rightarrow c_2$, where c_1 and c_2 are called abstract concept and reified concept, respectively, and "$\rightarrow$" is reification relationship. □

This strategy helps build a correct context for a reified concept. While a context of an abstract concept in MD is explicit and can be described by its parent concept, the context of a reified concept is, sometimes, implicit. For example, a concept such as "$25000" in Table 1 is difficult to be associated any concept if no mechanism disambiguates it. A pair-wise mechanism, described in Strategy 4, is a means of achieving it. For instance, if we have a paired concept (total, value((currency, USD), (amount, 25000)), we can easily derive the correct association of (total, USD25000), where USD25000 if a reification of "total" concept.

Strategy 5 (*Achieving sense making by concept matching*): Given $w_i \in$ SD and $c_j \in$ MD ($i, j \in 1 \ldots n$), MD makes sense against SD if and only if $\forall w_i, c_i, c_i \Rightarrow w_i$. □

This strategy ensures that all words in a structured tabular document can be transformed into the unique, explicit and independent concepts in a one-to-one match document. For example, the document of Table 1 can be transformed into another document shown in Table 3 (note: the definition can be replaced by a reference to a commonly shared vocabulary).

2.3 Document representation models

Employing the above strategies of SD-based document sense disambiguation, we can create two document representation models for representing both SD document and MD document. The former must satisfy the business user requirement for a tabular document presentation, while the latter opts for adapting to solve one-to-one match problem. With these in mind, our document representation models are designed as follows:

Definition 1 (*Document presentation model*): A document presentation model SD for business users is a tabular document model, defined as follows:

$$SD = (w_{11}, w_{12}, \ldots, w_{ij}, \ldots, w_{mn}),\ i, j \in 1...N \quad (3.1)$$

$$w_{ij} = (WID, \{FC\}, [GF]) \quad (3.2)$$

where SD is a tabular model, in which w_{ij} is a cell, i is a row number and j is a column number. Each cell is a tuple, in which WID is cell ID, {FC} is a set of words expressing w_{ij}, and [GF] is local graphic features. When a word $w_{ij} \neq$ NULL, the cell(w_{ij}) is active and implies a valid word, otherwise the null cell is considered as inactive and neglected. □

This tabular model well fits for the existing business practices for document template design and reification.

Definition 2 (*Document data model*): A document data model MD for storing one-to-one matching concepts of SD is a hierarchically structured document model, defined as follows:

$$MD = (c_1^1, (c_a, c_r[k])_i^2, ..., (c_a, c_r[k])_i^j, ..., (c_a, c_r[k])_i^n \quad (4.1)$$

$$c_a = (WID, \{FC\}, \{RT\}, PC, DP) \quad (4.2)$$

$$c_r = (RID, \{FC\}, \{RT\}, OP) \quad (4.3)$$

where MD is a hierarchical form, in which c_1^1 is document root, (c_a, $c_r[k]$) is a pair of an abstract concept and a set of reified concepts in k number, and i and j refer to any sibling concept and any MD level.

In each concept pair following Strategy 4, the abstract concept c_a is a tuple where WID is a unique document concept classifier extracted from SD, {FC} is a set of words of a WID-ed cell, {RT} is a set of unique concept identifiers one-to-one corresponding to {FC} and referenced in common vocabularies (VOC). DP = "yes/no" such that if yes {FC} is displayed else hide. PC points to the parent WID.

The reified concept c_r is also a tuple, in which RID, notated as WID-i pointing to WID. OP defines how {FC} and {RT} are related and presented in SD.

Document data model MD enable all data of SD document to be accurately and hierarchically represented, in which sense ambiguities are disambiguated by RT.

2.4 Transformation between SD and MD documents

SD and MD are two different models. The SD model is to provide a model that is similar to existing tabular document model while MD model is to provide a storage model. it is necessary to provide a tool to enable mutual transformation such that:

$$SD \Leftrightarrow MD \tag{5.1}$$

In this paper, a set of rules are designed to enable the transformation between SD and MD. These rules are provided as follows to govern the design of both SD and MD when models of Definition 1 and Definition 2 are followed:

Definition 3 (*Common concept reference rule*): While adopting *Strategy 1*, for any concept $c \in$ MD and any cell $w \in$ SD, there exist a set of unique concept definitions $an_i \Rightarrow rt_i \in$ VOC such that:

$$rt_i \Rightarrow fc_i \tag{6.1}$$

The (6.1) guarantees that for $\{rt_i\} \Rightarrow \{fc_i\} \Rightarrow c_j$ and $\{rt_i\} \Rightarrow \{fc_i\} \Rightarrow w_j$ hence $c_j = w_j$ of Strategy 1, a group c and a cell w share a set of unique and consistent word senses by rt.

Definition 4 (*Common group concept classifier rule*): while adopting *Strategy 2*, for any concept $c \in$ MD and cell $w \in$ SD, there exists a common group concept classifier *wid*, such that:

$$(wid \in c) = (wid \in w) \tag{7.1}$$

The (7.1) guarantees that the sense of w and c are equivalent by *wid* and thus transformable. In addition, it also solves one presentation problem of SD, that is, how to properly present w in tabular form. The solution is to utilize wid as hierarchical classifier such that wid = wid(wid). For example, if there are cells with classifiers 1.5.8.1, 1.5.9.2, 1.5.9.2-3, 1.5.8.1-2 and 1.6.8, then words of 1.5.8.1, 1.5.9.2, 1.5.9.2-3 and 1.5.8.1-2 is a group having a common root concept 1.5. Within this 1.5 group concept, 1.5.8.1-2 is the second value (reified concept) of 1.5.8.1, and 1.5.9.2-3 is the third value (reified concept) of 1.5.9.2. By this concept grouping rule, all words can be properly aligned in a tabular form such that a group of siblings can be placed in the same row or column and their value can be placed just under or right of column cells or row cells. The detailed presentation approach for SD is beyond the discussion of this paper will be elaborated elsewhere.

2.5 Procedures of Achieving $|A(i)_S| = 1$

Procedurally, the semantic consistency of the transformation from SD to MD can be validated through a semantic consistency check procedure as follows.

Procedure 1 (*Sense Consistency Check*): For any $id_i = (wid_i \mid rid_j) \in$ SD ($i \in 0 \ldots m$) and $id_j = (wid_j \mid rid_j) \in$ MD ($j \in 0 \ldots n$), a sense consistency check can be made to semantically validate both SD and MD, as defined in the following:

```
(1)  count = 0;
(2)  for(i=0, i<m, i++)
(3)  {
(4)        if (id_i = Search(id_j, MD))
(5)              MD = MD – id_j
(6)        else
(7)        {
(8)              InconsistentSD = count + 1;
```

```
(9)                         LogSD[InconsistentSD -1] = id_i;
(10)            }
(11)    }
(12)    InconsistentMD = Count(MD);
(13)    LogMD[InconsistentMD -1] = Enumerate(MD);
(14)    Inconsistent = InconsistentSD + InconsistentMD;
```

This procedure validates the sense consistency between cell semantics of SD and group concepts of MD. It provides the information about the sense inconsistent status. The InconsistentSD means that the sense inconsistency is caused by SD document while the InconsistentMD means that the sense inconsistency comes from MD document. Besides the above procedure, for each cell of SD, its corresponding group concept must also be checked whether $rt_i \Rightarrow fc_i$ exists. This is the task of Procedure 2.

Given the sense consistent SD and MD, when a recipient receives an MD document from the sender, this MD document must be semantically validated again to disambiguate document senses. We provide a sense existence match procedure to semantically validate an incoming document.

Procedure 2 (*Sense Existence Match*): For any incoming document MD such that $rt_i \in$ MD, $iid \in$ dbVOC, $i \in (1 \ldots m)$, a semantic validation is defined as follows:

```
(1)    count = 0;
(2)    for(i = 0, i < m, i++)
(3)    {
(4)        Search(iid, dbVOC);
(5)        {
(6)            if rt_i = iid
(7)                Continue()
(8)            else
(9)            {
(10)               NonExistent = count + 1;
(11)               log[NonExistent -1] = rt_i;    }    }    }
```

The above defined procedure semantically validates the incoming MD document from a trading partner. If NonExistent = 0, the MD document is 100% consistent for interpretation as the sender's meaning understanding. In this case, $| A(i)_S | = 1$ has been achieved. The interpretation accuracy is reduced or interpretation ambiguity is increased as number of NonExistent increases.

3 XPM Implementation of Document Data Model

The key to achieving $| A(i)_S | = 1$ is to derive a semantic consistent MD from SD. In this paper, the MD document designed in Section 2 is implemented in an XPM document model, which is initiated in the research of [5, 17]. The original XPM model describes a document as a set of hierarchical concepts, in which each concept is denoted as a denotation by a set of elementary structures. Every denotation is an independent concept. Several concepts are grouped by connotation such that a sequence of sibling denotations is connotations of their parent concept's denotation. Simply, XPM document model can be defined as follows:

```
<!ELEMENT concept (#PCDATA | concept)*>
<!ATTLIST concept
```

```
        iid ID #REQUIRED
        an CDATA #REQUIRED
>
```

In this document model, a concept is denoted by *iid* and *an*, and is connoted by (#PCDATA | concept)* in which each concept is recursive for deriving a hierarchy.

The simple structure of XPM is its advantage to combat complexity of the existing business world. Its denotation and connotation relationship enables this specification to separate all description of a concept (i.e. denotation) into an <ATTLIST> without the need of associating with other concept as described by an <ELEMENT>. Concept grouping (i.e. connotation) as a composite concept is achieved by the sub-hierarchy of any <ELEMENT>.

MD document model designed in this paper maintains these good features, which can be described as follows:

```
<!ELEMENT doc (concept*)>
<!ELEMENT concept (concept | values)*>
<!ELEMENT values (value*)>
<!ELEMENT value (#PCDATA)>
<!ATTLIST doc       cid CDATA #FIXED "d" rt CDATA #REQUIRED fc CDATA #REQUIRED
                    comment CDATA #IMPLIED>
<!ATTLIST concept  cid ID #REQUIRED  rt CDATA #REQUIRED fc CDATA #REQUIRED
            pc CDATA #REQUIRED dp (yes | no) "yes" oc CDATA #IMPLIED>
<!ATTLIST values    rid ID #REQUIRED  oc CDATA #IMPLIED ch CDATA #REQUIRED>
<!ATTLIST value     rid ID #REQUIRED  rt CDATA #IMPLIED op (EQ | LessThan | LargerThan |
LessEq | LargerEq | PlusMinus | IS | NOT | MF | FN) "IS"
            dt (string | number | decimal | scientific | const | serial | symbol) "string">
```

Figure 1: XPM MD Document Data Model (mddoc.dtd)

Applying the model of Figure 1, the MD1 document of Table 3 following SD1 of Table 1 can be instantiated as a reified XPM MD document, as shown in Figure 2.

```
<!DOCTYPE doc SYSTEM "mddoc.dtd">
<doc cid="d" rt="100" fc="Offer Sheet" comment="An offer sheet answering an inquiry sheet.">
    <concept cid="d.1" rt="100;;200" pc="d" fc="Offer No." dp="yes">
      <values rid="d.1-v"><value rid="d1.1-v1" dt="serial" op="IS">S111</value></values>
    </concept>
    <concept cid="d.2" rt="400" pc="d" fc="Commodity" dp="yes">
      <values rid="d.2-v"><value rid="d.2-v1" rt="500" dt="string" op="IS">Fridge</value></values>
    </concept>
    <concept    cid="d.3" rt="600" pc="d" fc="Description" dp="yes">
        <values rid="d.3-v" oc="3">
            <value rid="d.3.-v1" rt="700" dt="string" op="IS">orange</value>
            <value rid="d.3.-v2" dt="symbol" op="IS">,</value>
            <value rid="d.3-v3" rt="800;;900" dt="string" op="IS">low temperature</value>
        </values></concept>
    <concept cid="d.4" rt="1000"    pc="d"    fc="Quantity" dp="yes">
        <values rid="d.4-v" oc="2" ch="all">
            <value rid="d.4-v1" dt="number" op="EQ">100</value>
            <value rid="d.4-v2" rt="1200"    dt="string" op="IS">pieces</value>
        </values></concept>
    <concept cid="d.5" rt="1300" fc="price" pc="d" dp="yes">
        <values rid="d.5-v" oc="4" ch="all">
            <value rid="d.5-v1" rt="1400" dt="const" op="IS">$</value>
            <value rid="d.5-v2" dt="decimal" op="EQ">250.00</value>
            <value rid="d.5-v3" dt="symbol" op="IS">/</value>
```

```
            <value rid="d.5-v4" rt="1200" dt="string" op="IS">piece</value>
        </values></concept>
    <concept cid="d.6" rt="1600" fc="total" pc="d" dp="yes">
        <values rid="d.6-v" oc="2" ch="all">
            <value rid="d.6-v1" rt="1400" dt="const" op="IS">$</value>
            <value rid="d.6-v2" dt="decimal" op="EQ">25,000.00</value>
        </values></concept>
</doc>
```

Figure 2: XPM MD1 Reified Document (md1.xml)

The above XPM MD1 reified document is semantically consistent with SD1 document. It can be sent for further semantic match procedure of Procedure 2 for final sense disambiguation.

The example implementation attempts to demonstrate that how a tabular document can be semantically and consistently represented in a hierarchical document, where any of its sub-documents can be extracted and reused by referring to the root concept of the subdocument with the document name with URL link as the namespace. This type of interpretable semantic reuse is often difficult for normal tabular document.

4 Evaluation Method of Sense Interpretation Accuracy

The accuracy of SD document interpretation can be evaluated using a newly devised evaluation method, which takes consideration of interpretation inaccuracy from both the transformation of SD⇒MD and the sense match between MD and dbVOC.

Definition 5 (*Transformation Inaccuracy TI between SD and MD*): When an SD document is transformed to an MD document, the transformation inaccuracy TI happens such that for $wid_i \in$SD and $wid_j \in$MD, $wid_i \neq wid_j$, $i \in 0\ldots m$, $j \in 0\ldots n$. The number of TI can be calculated from N_{TI} = Inconsistent of Procedure 1.

Definition 6 (*Match Inaccuracy MI between MD and Vocabulary dbVOC*): When all the concepts of an MD document are compared with the concepts of a vocabulary dbVOV, the sense match inaccuracy MI happens such that $rt_i \in$MD and $iid_j \in$ dbVOC, $rt_i \neq iid_j$, $i \in 0\ldots m$, $j \in 0\ldots n$. The number of MI can be calculated from N_{MI} = NonExistent of Procedure 2.

Definition 7 (*Sense Interpretation Accuracy SIA*): The total sense interpretation accuracy of an MD, where $rt_i \in$ MD ($i \in 1\ldots n$), during achieving $|\ A(i)_S\ | = 1$, can be measured as follows:

$$SLA = \frac{T - N_{TI} - N_{MI}}{T} \tag{8.1}$$

where $T = Count(\sum_{i=1}^{n} rt_i)$, and N_{TI} and N_{MI} are defined in Definition 5 and 6, respectively.

The evaluation form SIA is useful to evaluate the sense making of any incoming MD document. Against SIA formula, the evaluation result of the above incoming offer sheet (md1.xml) is the 100% sense interpretation accuracy since both N_{TI} and

N_{MI} are zero such that SIA = 1 = 100%. It must be noted that a comparative evaluation against existing other approaches such as ontology will not be discussed here. The reason is that ontology and the vocabulary used within this paper has substantial difference. The vocabulary used in this paper is collaboratively created and is cross-domain [5, 6] while Ontology is domain-wide. This is also why SIA can equal 1.

5 Related Work

The SD-DSD approach in general, relates to the sense disambiguation technology [7, 11], which is often applied in the area of machine translation, information retrieval, hypertext navigation, content and thematic analysis, grammatical analysis, speech processing, and text processing. The general steps of achieving sense disambiguation are: (1) determining all the different senses for every relevant word in a text, and (2) assigning an appropriate sense to each occurrence of the word in a text.

The methods of taking the first step often require a list of predefined senses in sorts of vocabularies such as taxonomy (e.g., taxonomystrategies.com), thesaurus [1] (e.g., visualthesaurus.com), glossary (e.g., [9] and glossary.com), machine-readable dictionary (e.g. WordNet at wordnet.princeton.edu), ontology (e.g., [2] and geneontology.org), and collaborative concepts [6]. These vocabularies define different senses of each word that possibly occurs in a document needing disambiguation. The methods for Step (2) can be classified as context-based and knowledge-based [7] or unsupervised and supervised [11]. In practice, context and knowledge are both utilized in assigning the sense to words.

Particularly, SD-DSD approach is context- and knowledge-based [7]. In designing SD-DSD approach, different contexts of creating SD and MD documents are aligned and mediated through a collaborative editing system that links to an external common vocabulary exactly known by SD creator and MD interpreter. Through this editing system, a unique sense of any word is assigned to both SD and MD documents without sense ambiguity. It is achieved by applying a set of collaborative concepts [5, 6], i.e. CONEX concepts, which are mutually agreed in meaning by document senders and receivers, assuming both of them are in a same e-marketplace.

6 Conclusion

This paper achieves the meaning understanding of exchanged business documents between unknown sellers and buyers in e-marketplace by proposing a novel structured tabular document sense disambiguation approach (SD-DSD). In this approach, it has developed five sense disambiguation strategies, which guide designing two document representation models: structured tabular document (SD) presentation model, and one-to-one match document (MD) data model. Two procedures of sense consistency check procedure and sense existence match are developed to control the sense disambiguation. The MD document data model is implemented in a newly devised XPM scheme, where example of its reification is described. The interpretation

accuracy between SD creator and MD user is validated, considering both factors in SD-MD transformation and MD interpretation.

This paper has contributed a novel approach to tabular document sense disambiguation, achieving a unique sense of $| A(i)_S | = 1$. It has eased the difficult problem in word sense disambiguation (WSD) area by strategies of sense uniqueness, sense explicitization, sense independence, sense reification and sense making, and made the problem-solving possible.

SD-DSD approach is fundamental and important to e-marketplace semantic information exchange. Future work will use this research result for designing tools and applications of business document interchange.

References

1. Aitchison, J. and S. D. Clarke (2004) The Thesaurus: A Historical Viewpoint, with a Look to the Future. *Cataloging & Classification Quarterly* 37(3/4)5-21.
2. Gruber, T. R. (1993) A translation approach to portable ontologies. *Knowledge Acquisition* 5(2) 199-220.
3. Guo, J. (2007) Business-to-Business Electronic Marketplace Selection. *Enterprise Information Systems* 1(4) 383-419.
4. Guo, J. (2007) A Term in Search of the Infrastructure of Electronic Markets. In: *Research and Practical Issues of Enterprise Information Systems II Vol. 2*, IFIP Vol. 255:831-840.
5. Guo, J. (2008) *Collaborative Concept Exchange*, VDM Publishing, Germany.
6. Guo, J. (2009) Collaborative Conceptualization: Towards a Conceptual Foundation of Interoperable Electronic Product Catalogue System Design. *Enterprise Information Systems* 3(1), pp. 59-94.
7. Ide, N. and J. Véronis （1998） Introduction to the special issue on word sense disambiguation: the state of the art. *Computational Linguistics* 24(1) 2-40.
8. Kaleta, M., Pałka, P., Toczyłowski, E. and T. Traczyk (2009) Electronic Trading on Electricity Markets within a Multi-agent Framework. In: *Proc. of ICCCI 2009*, LNAI 5796, pp. 788–799.
9. Malkin, G. (1986) *RFC1983: Internet Users' Glossary*, RFC Editor, USA.
10. Michalak, T., Tyrowicz, J., McBurney, P. and M. Wooldridge (2009) Exogenous coalition formation in the e-marketplace based on geographical proximity. *Electronic Commerce Research and Applications* 8(4) 203-223.
11. NavigliI, R. (2009) Word Sense Disambiguation: A Survey. *ACM Computing Surveys* 41(2) Article 10.
12. Ncho, A. and E. Aimeur (2004) Building a Multi-Agent System for Automatic Negotiation in Web Service Applications. In *ACM Proc. of AAMAS'04*, pp. 1464-1465.
13. Serban, C., Chen, Y., Zhang, W. and N. Minsky (2008) The concept of decentralized and secure electronic marketplace. *Electronic Commerce Research* 8(1-2) 79-101.
14. Stohr, E. A and J. L. Zhao (2005) Workflow Automation: Overview and Research Issues. *Information Systems Frontiers* 3(3) 281–296.
15. Unitt, M. and I. C. Jones (1999) EDI - The Grand Daddy of Electronic Commerce. *BT Technology Journal* 17(3) 17-23.
16. Wang, S., Zheng, S., Xu, L., Li, D. and H. Meng (2008) A literature review of electronic marketplace research: Themes, theories and an integrative framework. *Information Systems Frontiers* 10(5) 555-571.
17. XPM, http://www.sftw.umac.mo/~jzguo/pages/resource.html.