Instytut Badań Systemowych
Polskiej Akademii Nauk

mgr inż. Paweł Maciej Szmeja

Autoreferat rozprawy doktorskiej

Wyznaczanie wielowymiarowego podobieństwa semantycznego w ontologiach

(Calculating multidimensional semantic similarity in ontologies)

Opieka naukowa:
dr hab. Maria Ganzha prof. PAN
dr Wiesław Pawłowski

Warszawa, 2020

Spis treści

1 Wstęp

1.1 Motywacja

Istnieją różne sposoby na mierzenie podobieństwa, z których wiele sięga poza matematykę i informatykę, zagłębiając się w psychologię, socjologię, a nawet filozofię. Powszechność i powszedniość problemu podobieństwa były i są inspiracją dla naukowców, którzy zaproponowali wiele podejść i metod obliczania podobieństwa, o różnym stopniu formalizacji.

W przeszłości, ze względu na swoje filozoficzne korzenie, pojęcie podobieństwa semantycznego odnosiło się do bliskości *znaczenia słów*. W miarę upływu lat to pojęcie zostało rozszerzone o podobieństwo zdań, dokumentów, a nawet całych zbiorów dokumentów. W tym obszarze, jako podstawa obliczeń i źródło wiedzy na temat języka naturalnego, używane były słowniki i tezaurusy. Ontologie, będące naturalnym rozszerzeniem idei słowników, były również coraz częściej wykorzystywane, w miarę wzrostu ich ogólnej popularności w informatyce. Dziś, ontologie i tezaurusy, takie jak WordNet [17] są często używane do obliczania podobieństwa słów, zdań i dokumentów. Inne, specyficzne dla konkretnych obszarów (domen) ontologie, takie jak ontologia genów [2] oferują przestrzeń, w której obliczanie podobieństwa semantycznego jest niezależne od znaczenia słów, a raczej operuje na wysoce wyspecjalizowanej wiedzy przechowywanej w ontologii. Rozpowszechnienie ontologii rozszerzyło znaczenie podobieństwa semantycznego (poczynając od *znaczenia słów*), o znaczenie *bytów i obiektów ontologicznych*.

Nowy, praktyczny, wymiar został nadany semantyce wraz z rozwojem narzędzi informatycznych takich, jak RDF, SPARQL, OWL, triplestores, a także podejść i idei w rodzaju Semantic Web, Linked Open Data, semantic data lakes, itd. Wraz z powstaniem technologii, które w praktyczny sposób umożliwiają semantyczną reprezentację i przetwarzanie informacji, problem podobieństwa zaczął być rozpatrywany w nowym, informatycznym, kontekście. Przechodząc od teorii do praktyki, można zaobserwować, że wraz z rosnącą liczbą semantycznie opisanych („adnotowanych") zasobów, rośnie potrzeba opracowania nowych metod i algorytmów przetwarzania takich danych. Jedną z najważniejszych właściwości danych semantycznych jest możliwość łączenia wielu heterogenicznych zasobów. Obliczanie podobieństwa jest co najmniej użyteczne, a czasami nawet niezbędne w procesach wyszukiwania wzajemnych powiązań, wnioskowaniu na podstawie istniejących informacji, lub, pisząc bardziej ogólnie, zautomatyzowanej interakcji z danymi semantycznymi.

Obliczanie podobieństwa bytów w ontologiach umożliwia wyszukiwanie, filtrowanie i inteligentne przetwarzanie semantycznie adnotowanych danych. Jest też fundamentem dla wielu inteligentnych aplikacji, algorytmów i rozwiązań dla różnorakich form zarządzania danymi, które coraz częściej korzystają z semantyki.

Pomimo znaczącego rozwoju technologii semantycznych, ogólne metody obliczania podobieństwa (tzw. miary podobieństwa) pozostają nieco w tyle. Analiza stanu wiedzy ujawnia szereg problemów z obecnie dostępnymi metodami obliczania podobieństwa w ramach ontologii. Metody, które bardzo głęboko analizują wiedzę ontologiczną są wysoce wyspecjalizowane i dostępne wyłącznie dla bardzo specyficznych ontologii domenowych. Ponieważ zakładają one określoną strukturę ontologii lub istnienie określonych właściwości i adnotacji, nie mogą być używane w dowolnych ontologiach. Z drugiej strony, metody, które są stosowalne dla dowolnych ontologii, często zdefiniowane na wysokim poziomie abstrakcji, nigdy nie wykorzystują pełnego zakresu dostępnej wiedzy (lub ekspresywności logicznej), bardzo często ograniczając się do taksonomii. Te i inne ograniczenia sugerują możliwość poprawy stanu wiedzy, niekoniecznie jeśli chodzi o wydajność implementacji

oprogramowania, ale raczej lepsze wykorzystanie wiedzy zawartej w ontologiach.

W obszarach, takich jak Internet Rzeczy (IoT), dane, które często są najbardziej interesujące dla badaczy i najważniejsze dla użytkowników, to dane „transakcyjne" (na przykład dane obserwacji lub akcji, które tracą znaczenie w czasie), lub dotyczą instancji (na przykład informacje o konkretnych sensorach). Tradycyjne miary podobieństwa, ze względu na swoje skupienie na taksonomii, mają ograniczoną użyteczność w wielu zastosowaniach IoT, ponieważ charakteryzują się niewrażliwością na zmiany w danych transakcyjnych.

Kolejnym problemem jest domniemana uniwersalność wyniku obliczania podobieństwa. Różne metody przedstawiają różne, nawet w skrajnym stopniu, wyniki. Stosując różne metody można spotkać się z sytuacją, w której jedna metoda nie rozpoznaje podobieństwa w ogóle, a inne stwierdzają znaczące podobieństwo. Pomimo tego faktu, mówi się, że wszystkie miary obliczają „to samo" *semantyczne* podobieństwo, które funkcjonuje jako pojęcie uniwersalne, stosowalne do dowolnego wyniku dowolnej miary podobieństwa. Mimo różnego rozumienia i sposobu obliczania podobieństwa proponowanych przez różnych autorów, wydaje się, że celem jest zawsze odwzorowanie jakiegoś teoretycznego, idealnego podobieństwa. Wynik podobieństwa nie ma więc dodatkowej interpretacji i nie odzwierciedla różnicy w podejściach, a jedyną informacją (ewentualnie) dołączoną do wyniku jest nazwa użytej miary.

1.2 Cele badawcze

W świetle wyżej wymienionych zagadnień, związanych z interpretacją wyników podobieństwa i powstałych dotychczas współczesnych miar, potrzebne jest uaktualnione i zmodernizowane podejście do obliczania podobieństwa, w tym podobieństwa w ontologii. Wobec tego zdefiniowano następujące cele badawcze:

1. Stworzenie sposobu opisu podobieństwa niezależnego od dotychczasowych podejść do klasyfikacji i grupowania modeli i miar podobieństwa. Podobieństwo powinno mieć, jasną interpretację i znaczenie obejmujące dotychczas stosowane miary. Podejście powinno być stosowalne do wyników metod obecnie istniejących, jak i tych utworzonych w przyszłości.

2. Definicja modelu podobieństwa, zawierającego ogólny algorytm wyznaczania podobieństwa semantycznego, stosowalnego niezależnie od domeny, struktury danych, lub sposobu reprezentacji wiedzy, i zdolnego do generalizacji istniejących metod.

3. Definicja i implementacja ogólnego algorytmu wyznaczania podobieństwa semantycznego, razem z przykładami dla języka OWL.

1.3 Podsumowanie zawartości rozprawy

Jako narzędzie do grupowania istniejących metod podobieństwa, w pracy zaproponowane zostało wymiarowe podejście do obliczania podobieństwa, wypełniające warunki stawiane w pierwszym celu badawczym. Przedstawia ono teoretyczny model podziału wiedzy na wymiary, w kontekście podobieństwa semantycznego. Wprowadzenie pojęcia wymiaru podobieństwa oferuje nowy kontekst w którym analizowane i porównywane mogą być różne metody. Dodatkowa informacja na temat „wymiarowości" podobieństwa nadaje wynikom wyraźną interpretację, według której wyniki

dostarczają informacji na temat różnych aspektów podobieństwa. Pozwala to na wyraźne rozróżnienie i wytłumaczenie rozbieżnych wyników różnych metod, oraz bardziej świadomy wybór metody obliczania podobieństwa dla dowolnego zadanego problemu. Opis podejścia wymiarowego znajduje się w sekcji 3.

Rozprawa przedstawia również wyniki badań nad podobieństwem semantycznym, ze szczególnym wskazaniem na podobieństwo w ontologiach. Oprócz analizy stanu wiedzy, głównym wkładem jest platforma („framework") podobieństwa *SimDim* zawierająca ogólny algorytm podobieństwa, będący realizacją nowego cechowego modelu podobieństwa (realizując tym samym drugi cel badawczy), oraz opis *wymiarowego* podejścia do podobieństwa, które jest realizowalne za pomocą wspomnianego algorytmu. Platforma *SimDim* jest w stanie włączyć w obliczanie podobieństwa wiedzę często pomijaną w innych podejściach do obliczania podobieństwa, a tym samym wzbogacenie aktualnego stanu wiedzy. *SimDim* oferuje wysoki stopień konfiguracji, a dobór parametrów konfiguracyjnych ma duży wpływ jej właściwości, zachowanie i wydajność.

Generyczny algorytm podobieństwa semantycznego wraz z implementacją i przykładami dla języka OWL, wchodzący w skład platformy *SimDim*, jest oparty na podejściu cechowym i realizuje trzeci cel badawczy. Ze względu na definicję na wysokim poziomie abstrakcji jest on stosowalny do dowolnej domeny, a nawet do danych spoza ontologii (choć rozprawa skupia się na podobieństwie w ontologiach). Dobranie różnych parametrów algorytmu zmienia jego zachowanie w znaczący sposób, a różne parametry tworzą osobne *instancje SimDim*, pozwalając na realizację w ramach *SimDim* innych istniejących podejść. Rozprawa zawiera również opis wielu przykładowych parametrów (będących funkcjami) i instancji. Algorytm i jego parametry są opisane w rozdziale 4. Częścią pracy jest implementacja algorytmu, wraz z przykładowymi instancjami *SimDim*.

2 Podobieństwo semantyczne – stan wiedzy

Poniższe sekcje przedstawiają podsumowanie analizy obecnego stanu wiedzy w zakresie podobieństwa semantycznego. Pojęcie podobieństwa było wielokrotnie analizowane i przedstawiane w zakresie różnorodnych, mniej lub bardziej sformalizowanych, podejść. Powstałe na przestrzeni lat miary podobieństwa wymykają się jednoznacznej klasyfikacji [18] i reprezentują bardzo różne punkty widzenia, z których podobieństwo jest analizowane. Ogólny konsensus dotyczący zdecydowanej większości miar podobieństwa jest ograniczony do ogólnych stwierdzeń. Jednymi z pierwszych stwierdzeń, z którymi zgodna jest znacząca ilość miar, są te wyznaczone przez Lin'a w [24]. We wspomnianej pracy Lin stwierdził, że podobieństwo dwóch dowolnych bytów powinno być, po pierwsze, proporcjonalne do ilości rzeczy, które są dla tych bytów wspólne, a po drugie, odwrotnie proporcjonalne do ilości różnic pomiędzy nimi. Dodatkowo, maksymalne podobieństwo oznacza, że dane byty są identyczne.

Te, i inne „intuicje" zostały wielokrotnie sformalizowane na różnych poziomach abstrakcji, i dla różnych domen, prowadząc do powstania różnorakich miar podobieństwa. Należy zauważyć, że niektóre modele podobieństwa są bardzo ogólne, i z tego względu szeroko stosowalne, podczas gdy inne są wyspecjalizowane i zakładają istnienie konkretnych struktur informacji np. atomowych cech bytów, struktury grafu, lub taksonomii. Poniżej zostały podsumowane najważniejsze istniejące modele i miary podobieństwa, oraz przykłady miar wyspecjalizowanych dla konkretnych domen.

2.1 Podstawowe miary podobieństwa

Jedną z najstarszych miar podobieństwa, która do dziś stanowi inspirację dla nowoczesnych podejść, jest tak zwany *indeks Jaccarda* [20]:

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} \tag{2.1}$$

Ta miara jest zdefiniowana dla dowolnych zbiorów X i Y. W podejściu Jaccarda, elementy tych zbiorów są nazywane *cechami*, które reprezentują byty, których podobieństwo obliczamy. Indeks Jaccarda to pierwsza miara, w której wprowadzona została idea reprezentacji bytów poprzez zbiory cech. Samo pojęcie cechy nie zostało formalnie zdefiniowane, a wyznaczenie cech (w teorii możliwe dla dowolnego bytu) zostało pozostawione w sferze intuicji. Podejście cechowe zostało wiele lat później rozwinięte przez Tverskiego, który w swojej pracy [46] zaproponował dwa modele podobieństwa, każdy reprezentowany przez jedną miarę:

Model ułamkowy Tversky'ego:

$$sim_{Tversky}(X,Y) = \frac{\alpha g(X \cap Y)}{\alpha g(X \cap Y) + \beta g(X \setminus Y) + \gamma g(Y \setminus X)} \tag{2.2}$$

Model kontrastowy Tversky'ego:

$$sim_{TverskyContrast}(X,Y) = \alpha g(X \cap Y) - \beta g(X \setminus Y) - \gamma g(Y \setminus X) \tag{2.3}$$

W powyższych równaniach g to funkcja liczbowa będąca skalą liniową, operująca na zbiorach, a α, β i γ to dodatnie parametry liczbowe (β i γ nie mogą wynosić zero). Obydwa modele są zbiorczo nazywane „modelem cechowym Tversky'ego". Podobnie, jak w pracy Jaccarda, X i Y reprezentują zbiory cech porównywanych bytów. Dla różnych parametrów, modele Tversky'ego wykazują różne właściwości. W szczególności, dla $\alpha = \beta = \gamma = 1$ i $g = |\cdot|$ (liczność zbioru) formuła modelu ułamkowego staje się identyczna z indeksem Jaccarda.

2.2 Podobieństwo w grafie

Modele grafowe (zwane też metodami krawędziowymi, lub odległości ścieżkowej) są stosowalne dla danych, które mogą być przedstawione w formie grafu (np. ontologii lub taksonomii). Przyjmują one założenie, że odległość węzłów w grafie (liczona według brzegów) ma znaczenie dla podobieństwa. Na wykresach z ważonymi krawędziami długość ścieżki jest sumą wag wzdłuż krawędzi, podczas gdy ścieżki nieważone upraszczają problem do liczenia krawędzi. Modele grafowe patrzą na ontologie, jak na skierowane grafy, gdzie małe odległości wzdłuż niektórych (nieważonych) typów krawędzi (np. w taksonomii) świadczy o podobieństwie, a długie ścieżki wskazują na odmienność (brak podobieństwa). Powszechną krytyką metod grafowych, w zastosowaniach ontologicznych, jest to, że pracują przy założeniu, że każda krawędź w ścieżce ma takie same znaczenie („wagę") dla semantyki. W praktyce jednak nie ma na to formalnych dowodów, a niektóre poszlaki, wskazują przeciwnie [3, 47].

Jedno z najprostszych rozwiązań dla metod grafowych (opublikowane w [34]) uzależnia podobieństwo od długości najkrótszej ścieżki między parą bytów (np. klas):

$$sim_{Rada}(c_1,c_2) = \frac{1}{\min(\text{paths}(c_1,c_2)) + 1} \tag{2.4}$$

W powyższym równaniu, paths(c_1, c_2) to zbiór odległości wszystkich ścieżek pomiędzy klasami c_1, a c_2 w taksonomii. Mimo tego, że oryginalna definicja zakłada istnienie taksonomii, można ją w trywialny sposób zastosować dla węzłów w dowolnym grafie. Bardziej wyrafinowane metody, takie jak zdefiniowana przez Wu w [48]:

$$Sim_{Wu}(c_1, c_2) = \frac{2 \cdot depth(LCS(c_1, c_2))}{depth(c_1) + depth(c_2)} \tag{2.5}$$

włączają normalizację i głębokość (i.e. odległość od korzenia) porównywanych bytów, głębokość LCS (Least Common Subsumer) porównywanych bytów, lub odległość ścieżek pomiędzy korzeniem, LCS, i samymi bytami.

Natomiast, podejście opisane w [37] używa LCS:

$$sim_{Resnik-edge}(c_1, c_2) = 2 \cdot depth(G) - \min(paths(c_1, LCS(c_1, c_2))) - \min(paths(c_2, LCS(c_1, c_2))) \tag{2.6}$$

Tutaj, $depth(G)$ oznacza głębokość grafu taksonomii (maksymalną odległość od korzenia do liścia).

2.3 Metody informacyjne

Metody informacyjne (bazujące na pojęciu „zawartości informacji") patrzą na podobieństwo z punktu widzenia teorii informacji [41] i przyjmują, że podobieństwo jest ściśle związane z ilością informacji zawartej w porównywanych bytach. Ta klasa metod jest reprezentowana przez model *zawartości informacji* („Information Content"; IC) zaproponowany przez Resnik'a w [37]. IC bytu x jest obliczana na podstawie jego *prawdopodobieństwa* ($p(x)$): $IC(x) = -\log(p(x))$, co sprawia, że dobrze nadaje się do stosowania w analizie tekstu, gdzie prawdopodobieństwo podmiotu tekstowego (słowo, zdanie, itp.) jest proporcjonalne do liczby wystąpień w tekście lub zbiorze dokumentów.

W kontekście ontologii prawdopodobieństwo wystąpienia bytu może być zdefiniowane na wiele sposobów. Metoda Resnik'a (ograniczona do taksonomii) określa, że prawdopodobieństwo bytu jest odwrotnie proporcjonalne do liczby klas, dla których jest on przodkiem. Według tej definicji, IC maleje monotonicznie od liści (największa zawartość informacji) do korzenia (najmniejsza zawartość informacji). Podobieństwo Resnik'a jest obliczane na podstawie z IC *najbardziej informatywnego wspólnego przodka* (Most Informative Common Ancestor; MICA) – przodka wspólnego dla porównywanych klas, który ma maksymalną IC:

$$sim_{Resnik}(c_1, c_2) = IC(MICA(c_1, c_2)) \tag{2.7}$$

MICA jest ściśle związany z LCS. Niektóre prace opierają się na podejściu Resnik'a, znosząc, lub osłabiając zależność od LCS. Na przykład, Lin zaproponował w [24] formułę, która obejmuje zawartość informacji samych bytów, a także ich LCS:

$$sim_{Lin}(c_1, c_2) = \frac{2 \cdot sim_{Resnik}(c_1, c_2)}{IC(c_1) + IC(c_2)} \tag{2.8}$$

Włączenia $IC(c_1)$ i $IC(c_2)$ dokonano w celu obejścia pewnej wady obecnej w modelu Resnik'a, gdzie wszystkie klasy, które dzielą $MICA(c_1, c_2)$ zawsze mają takie same podobieństwo, niezależnie od ich zawartości informacji. Podobne podejście doprowadziło do innej formuły zaproponowanej w [33]:

$$sim_{Pirro}(c_1, c_2) = \frac{sim_{Resnik}(c_1, c_2)}{IC(c_1) + IC(c_2) - sim_{Resnik}(c_1, c_2)} \tag{2.9}$$

Modyfikacja formuły Lin'a, opublikowana w [40], używa następującego prawdopodobieństwa wystąpienia $MICA(c_1, c_2)$:

$$sim_{Schlicker}(c_1, c_2) = sim_{Lin}(c_1, c_2) \cdot (1 - p(MICA(c_1, c_2))) \tag{2.10}$$

Nieco inne podejście zaproponowano w [28], gdzie podobieństwo jest proporcjonalne do sumy zawartości informacyjnej bytów:

$$sim_{Mazandu}(c_1, c_2) = \frac{\Sigma_{Z \in (A(c_1) \cap A(c_2))} IC(Z)}{\Sigma_{Z \in (A(c_1) \cup A(c_2))} IC(Z)} \tag{2.11}$$

W powyższym równaniu $A(c)$ oznacza zbiór przodków klasy c. Warto zauważyć, że to równanie jest bardzo podobne do indeksu Jaccard'a, który jest często używany jako inspiracja dla innych miar podobieństwa.

2.4 Inne miary podobieństwa

Metody opisane powyżej mogą być (często bezpośrednio) stosowane do obliczenia podobieństwa w ontologiach, w szczególności jeśli chodzi o podobieństwo klas. Starsze metody są często reinterpretowane i stanowią inspirację dla nowych miar. Dla przykładu, w [26] została przedstawiona prosta modyfikacja indeksu Jaccard dostosowująca go do klas w ontologii:

$$sim_{Maedche}(c_1, c_2) = \frac{|A(c_1) \cap A(c_2)|}{|A(c_1) \cup A(c_2)|} \tag{2.12}$$

gdzie $A(c_1)$ to zbiór przodków c_1. Natomiast w pracy [38] została zaproponowana podobna modyfikacja modelu ułamkowego Tversky'ego, także używająca zbioru przodków:

$$sim_{Rodriguez}(c_1, c_2) = \frac{|A(c_1) \cap A(c_2)|}{\gamma |A(c_1) \setminus A(c_2)| + (1 - \gamma)|A(c_2) \setminus A(c_1)| + |A(c_1) \cap A(c_2)|} \tag{2.13}$$

W zbliżony sposób niektóre metody podejścia cechowego wykorzystują zestaw instancji klasy jako zbiór jej cech, co zostało zilustrowane w [12]:

$$sim_{D'Amato}(c_1, c_2) = \frac{M}{L}\left(1 - \frac{L}{|I|}\left(1 - \frac{M}{L}\right)\right) \tag{2.14}$$

gdzie $M = min(|I(c_1)|, |I(c_2)|)$, $L = |I(LCS(c_1, c_2))|$, I to zbiór wszystkich instancji w ontologii, a $I(c) \subseteq I$ to zbiór instancji klasy c.

W związku z pokrewieństwem logiki opisowej z popularnym językiem zapisu ontologii OWL, miary zdefiniowane dla logiki opisowej są także ważne dla obliczania podobieństwa ontologicznego. Dla przykładu, w pracy [10] została zdefiniowana następująca miara dla logiki $\mathcal{ALC}$:

$$sim_{D'Amato\text{-}Instance}(c_1, c_2) = \frac{|(c_1 \sqcap c_2)^I|}{|c_1^I| + |c_2^I| - |(c_1 \sqcap c_2)^I|} \cdot max\left(\frac{|(c_1 \sqcap c_2)^I|}{|c_1^I|}, \frac{|(c_1 \sqcap c_2)^I|}{|c_2^I|}\right) \tag{2.15}$$

W powyższym równaniu $\_^I$ oznacza „rozwinięcie konceptu", które w terminologii ontologicznej jest równoważne ze zbiorem instancji danej klasy. Δ^I to zbiór wszystkich instancji, a GCS oznacza *Good Common Subsumer*, czyli pojęcie równoważne z *LCS*. Użycie ułamków i zbiorów do obliczania podobieństwa jest w tym podejściu inspirowane podejściami Tversky'ego i Jaccarda, co przyznają sami autorzy. Mimo tych inspiracji, ta, jak i inne podobne metody, nie są klasyfikowane, jako należące do podejścia cechowego, co wskazuje na trudności w jednoznacznej klasyfikacji i konieczność podejmowania arbitralnych decyzji przy organizacji miar w grupy.

Istnieje wiele różnych metod realizujących podejście cechowe, w tym te opisane w rozprawie, i inne podsumowane w [18], z których wiele przedstawia cechy semantyczne jako elementy zbiorów i często używa funkcji liczności zbioru. Różnią się one jeśli chodzi o definicję samych cech, gdzie popularne są zbiory przodków lub instancji. W nowoczesnych metodach zbiory cech są *ostre* – czyli każdy element jest cechą jednostki, lub nie, i *atomowe* – każda funkcja jest reprezentowana wyłącznie przez pojedynczy symbol, a głębsze znaczenie symbolu nie jest rozważane przy porównywaniu zbiorów.

Mimo znaczących pokrewieństw, różne podejścia do wyznaczania podobieństwa semantycznego wymykają się łatwej klasyfikacji. Na przykład, można rozważyć metody zawartości informacji, które używają IC przodków określonej klasy jako metody podobieństwa cechowego, gdzie zbiór cech składa się z przodków. Tutaj różnica w klasyfikacji metody jest bardzo mała. Niektórzy autorzy metod IC, np. [7, 8, 9] wykorzystują zbiory rozmyte, gdzie stopień przynależności jest proporcjonalny do zawartości informacji. Mimo że użyty został cechowy model Tversky'ego, ich autorzy uważają, że ich metody są informacyjne, a nie cechowe.

Inne sposoby klasyfikowania modeli podobieństwa zostały opisane w [18] dla metod przetwarzania języka naturalnego (NLP) lub w [42], dla metod odwzorowywania ontologii (tzw. „matching"). Jak stwierdzono w [18] modele podobieństwa nie poddają się łatwo sztywnej klasyfikacji. Niektóre metody (nazywane *hybrydowymi*) używają podejść inspirowanych więcej niż jednym modelem, lub metodą podobieństwa, podczas gdy inne mogą zmienić własne zaklasyfikowanie, poprzez użycie niewielkich modyfikacji.

Pełna rozprawa zawiera opis większej ilości miar podobieństwa, jak również podsumowanie różnych podejść do ich klasyfikacji.

3 Wymiary podobieństwa

Poniższa sekcja podsumowuje wymiarowe podejście do obliczania podobieństwa semantycznego, w szczegółach opisane w [44] i [45] i rozwinięte w pełni w rozprawie.

Wchodząc na chwilę na wyższy poziom abstrakcji przypomnijmy, że podobieństwo i znaczenie (semantyka) to z natury ludzkie i intuicyjne koncepcje. Z tego punktu widzenia wynik oceny podobieństwa powinien mieć wyjaśnienie (lub interpretację) zrozumiałą dla człowieka. Rozważmy prosty przykład porównania dwóch obiektów fizycznych: istnieje wiele *sposobów*, na które obiekty mogą być podobne lub odmienne – dwa z nich to kształt i kolor. Te dwa rodzaje cech są niezależne w odniesieniu do podobieństwa, tzn. obiekty mogą mieć podobny kolor i inny kształt (lub odwrotnie). W tej sytuacji standardowe sposoby automatycznego obliczania podobieństwa przedstawiają jeden wynik, który ma w jakiś sposób połączyć podobieństwo kształtu i koloru. Jednak te dwa podobieństwa, liczone oddzielnie, dostarczają więcej informacji, ponieważ mają one jasną *interpretację*. Dlatego możemy założyć, że osoba, która zna tę interpretację, ma lepsze zrozumienie

tego, jak podobne są dwa obiekty. W tym prostym przykładzie, kształt i kolor przynależą do dwóch oddzielnych *wymiarów* podobieństwa.

Można wysnuć z tego przypuszczenie, że podobieństwo bytów semantycznych ma wiele różnych aspektów, które zwykle są grupowane razem, w oparciu o to, jaka część dostępnych danych (lub wiedzy) jest używana (niezależnie od użytej metody, czy funkcji obliczania podobieństwa). Grupy te reprezentują różne typy (wymiary) relacji semantycznych, a zatem różne podobieństwo.

Podchodząc do tego zagadnienia z innej perspektywy, należy zauważyć, że współczesne sposoby grupowania atrybutów (funkcje) lub podziału danych na grupy (tu zwane wymiarami) koncentrowały się na wynikach podanych przez znane z góry metody. Innymi słowy, *punktem wyjścia dla klasyfikacji była metoda*, która dostarczała podstaw do interpretacji wyniku. Na przykład, podobieństwo semantyczne było opisywane, jako „podobieństwo Resnik'a", „podobieństwo Wu" itd.

Zaproponowane w ramach prowadzonych badań podejście wymiarowe zaczyna od wyjaśnienia *natury i znaczenia* wymiaru wiedzy, którym jesteśmy zainteresowani, a dopiero potem proponuje metody, które są wstanie dostarczyć wynik w tym właśnie wymiarze. W ten sposób rozpoznawane wymiary są wiedzione *interpretacją* wyniku, a nie użytą metodą. Co więcej, ta sama metoda może być stosowana w różnych wymiarach, co zilustrowano w następnych sekcjach i na rozbudowanym przykładzie opisanym w pełnej rozprawie.

Koncepcja różnych rodzajów podobieństwa była obecna w literaturze, w takiej czy innej formie, przez długi czas. Dla przykładu, [15] zawiera podsumowanie metod odwzorowania ontologii i klasyfikuje je według rodzaju danych, których używają. Przykładowe „rodzaje" metod używają porównania etykiet bytów, ich „atrybutów", instancji klas, położenia klas w taksonomii i innych. Kategoryzacja opisana we wspomnianej pracy uzupełnia bardziej ogólne prace nad odwzorowywaniem schematów [35], które prezentują własny podział metod odwzorowujących, według typu.

Późniejsza praca [14] podsumowuje metody odwzorowania ontologii i rozróżnia je na takie, które używają struktury ontologii i takie, które wykorzystują instancje (nazywając je kolejno wymiarami *strukturalnymi* i *jednostkowymi*). Rozważania kategoryzacja wykracza dalej wraz z wymiarami takimi, jak *składniowy*, *semantyczny*, *zewnętrzny*, *terminologiczny*, *rozszerzeniowy* i inne, z których niektóre nakładają się (jest to szczegółowo wyjaśnione w [14]).

Stan wiedzy dotyczącej odwzorowywania ontologii (ang. „matching" lub „alignment"), tematu bardzo pokrewnego do obliczania podobieństwa i korzystającego z miar podobieństwa, przedstawiony w [16], zawiera bardziej szczegółowe opisy różnych sposobów odwzorowywania ontologii z konkretnymi przykładami implementacji. Późniejsza praca [42], proponuje nieco inny podział na metody językowe, lingwistyczne, strukturalne i metody ciągu znaków (ang. „string–based").

Jeszcze innym przykładem jest praca [24], która wspomina, że różne miary podobieństwa mają różne ukryte założenia, co może wskazywać na istnienie wymiarów podobieństwa. Dla ontologii genów [2] są tam zdefiniowane dwa rodzaje miar podobieństwa, mianowicie dla par i dla grup, które są zbliżone do rodzajów podobieństwa, choć specyficzne tylko dla tej ontologii.

Kolejna praca na temat podobieństwa semantycznego [32], klasyfikuje istniejące metody dla ontologii biomedycznych w odniesieniu do zakresu (jakie podmioty są brane pod uwagę), źródła danych (krawędzie, węzły lub inne) i metryki (użyte funkcje odległości). Autorzy tej pracy przestrzegają, że wymienione metody, wykorzystują różne metryki i dane, oraz dają różne wyniki, ale mimo tego wszystkie podają, wynik podobieństwa, który jest, jak twierdzą, „uniwersalny".

Warto również wspomnieć o zaimplementowanym systemie odwzorowywania ontologii ASMOV [22], który definiuje i używa czterech wymiarów (*leksykalnego*, *relacyjnego*, *wewnętrznego* i *rozszerzeniowego*). Wyniki z poszczególnych wymiarów są ważone i sumowane w celu uzyskania ostatecz-

nego rezultatu.

Ważne jest, aby uświadomić sobie, że najbardziej „optymalne" wymiary podobieństwa powinny stanowić ortogonalny podział „całego" podobieństwa. Ponieważ, jak wspomniano wcześniej, rozróżnienie polega na rodzaju użytych danych, dostępną wiedzę należy podzielić na podzbiory, po jednym dla każdego wymiaru. Zgodnie z tą właściwością, wyniki podobieństwa dla każdego wymiaru byłyby niezależne. Należy tutaj jednak podkreślić, że w praktyce taki wyraźny (i ortogonalny) podział najczęściej jest niemożliwy.

W tym kontekście poniższe sekcje wprowadzają wybrane wymiary podobieństwa dla ogólnego przypadku porównania par bytów w ontologii.

3.1 Wymiar taksonomiczny

Podobieństwo w wymiarze *taksonomicznym* (zwane również *hierarchicznym*) opisuje, jak podobne są byty, według ich typów, lub przynależności do klas. Podobieństwo taksonomiczne może być zdefiniowane w dowolnej domenie, w której możliwe jest stworzenie taksonomii. Podczas gdy znaczenie „typu" w przypadku ogólnym nie jest precyzyjnie zdefiniowane, ma za to bardzo konkretne interpretacje i formalne definicje w praktycznych podejściach do obliczania podobieństwa.

W ontologiach i logice opisowej wymiar *taksonomiczny* jest najprościej opisany w kategoriach *typów* pojęć (tj. przodków).Podobieństwo wzrasta wraz z każdym typem klasy wspólnym dla porównywanych bytów, i maleje wraz z ilością typów różnych (rozbieżnych).

W praktyce taksonomia jest często wizualizowana jako graf, gdzie węzły są klasami, a krawędzie są relacjami podklasowości. Ze względu na strukturę taksonomii w logice opisowej, każdy typ wspólny dla porównywanych bytów leży na pewnej ścieżce z klasy uniwersalnej (*top*) do jednego z bytów. Dokładniej mówiąc, wspólność jest definiowana przez dowolną ścieżkę do LCS obu podmiotów. Każda krawędź na takiej ścieżce znajduje się między dwoma wspólnymi typami. Wszelkie krawędzie od *top* do któregokolwiek z podmiotów, które nie znajdują się między *top* i LCS świadczą o odmienności (tzn. zmniejszają podobieństwo).

Istnieje wiele metod bazujących na wykorzystaniu odległości ścieżkowej (które używają taksonomicznych przodków, [31]), niektóre metody IC (jak [23] lub [27]) i metody cechowe (np. [19]) mogą także być używane w tym wymiarze.

3.2 Wymiar opisowy

Z teoretycznego punktu widzenia wymiar *opisowy* zajmuje się właściwościami, które dany byt „ma", w przeciwieństwie do tego, czym „jest" (co jest ujęte w wymiarze *taksonomicznym*). Ogólnie rzecz biorąc, wymiar *opisowy* zawiera w sobie atrybuty, lub właściwości bytów, często łączące ze sobą dwa byty poprzez jakąś relację.

Wymiar opisowy jest stosowalny w każdym przypadku, gdzie opis bytu może zostać podzielony na właściwości, które ten byt wykazuje. Dla instancji ontologicznych, asercje właściwości ich dotyczące należą do tego wymiaru. Na przykład dla dokumentów mogą to być np. informacje dotyczące autora, użytego języka, data ostatniej modyfikacji itd. W związku z pokrewieństwem właściwości, a wspomnianych wcześniej cech, rozumianych w kontekście metod cechowych (patrz rozdział 2.1), to właśnie te metody są naturalnie dobrze dopasowane do wymiaru opisowego, gdyż każda właściwość może być interpretowana, jako cecha.

W niektórych ontologiach, wyraźnie rozróżnienie między danymi *taksonomicznymi* i *opisowymi* może być problematyczne, jeśli chodzi o byty, które tworzą hierarchię. Różnica między tymi dwoma wymiarami i to, czy nakładają się lub są całkowicie ortogonalne, sprowadza się do sposobu w jaki hierarchia jest konstruowana przez inżyniera ontologii.

Formalnie, wymiar *opisowy* ontologii opisuje relacje, które są właściwościami i nie określają podstawowych relacji pomiędzy klasami (subsumpcja, supersumpcja, równoważność klas). W terminologii logiki opisowej takie relacje są albo asercjami właściwości (np. $r(a, b)$) w przypadku instancji, lub restrykcjami (np. $\exists p.C$, $\forall t.5$) w przypadku opisów klas.

Należy podkreślić, że, niestety, istniejące metody zwykle nie rozróżniają między *taksonomicznymi* a *opisowymi* danymi. Zamiast tego niejawnie zakładają, że każda restrykcja właściwości przyczynia się do położenia bytu w taksonomii i nie ma dodatkowego (tzn. oddzielnego) wpływu na podobieństwo. W związku z tym nie istnieją metody, które można jednoznacznie opisać, jako czysto *opisowe*.

3.3 Wymiar leksykalny

Metody *leksykalne* wykorzystują słowniki, tezaurusy i ontologie leksykalne do oceny podobieństwa bytów (np. [1]). Byty w tym wymiarze są często interpretowane niebezpośrednio, w kontekście zewnętrznego słownika lub tezaurusa (gdzie są one wyszukiwane przez identyfikator, zwany etykietą). Dowolną parę etykiet (np. *rdfs:label* w RDF i OWL) lub nazw jednostek napisanych w języku naturalnym można poddać metodom podobieństwa wymiaru *leksykalnego*. Same metody mogą być bardzo skomplikowane i korzystać z dużych ontologii, słowników, lub tezaurusów (takich jak WordNet lub tezaurus Roget'a [21]).

Wymiar *leksykalny* jest najbardziej użyteczny dla bytów, które mają znaczące i jednoznacznie identyfikujące nazwy lub etykiety. Jest naturalnie dopasowany do dokumentów pisanych w języku naturalnym, gdyż w tym kontekście byty (słowa, paragrafy itd) są dokładnie równoważne z tymi ze słownika. Sytuacja jest nieco odmienna, w przypadku ontologii, gdzie byty oznaczone etykietami mają również zdefiniowane i formalnie zapisane dodatkowe właściwości, które nie są brane pod uwagę w kontekście słownikowym.

W przypadku ontologii, lub innych zasobów wychodzących poza zakres języka naturalnego, metody *leksykalne* często wyodrębniają etykiety bytów i używają np. zbiorów synonimów z WordNet (*synsetów*) jako zbiorów cech w metodzie cechowej. Wymaga to istnienia etykiet, które jednoznacznie identyfikują dany byt. W związku z tym metody leksykalne są wrażliwe na wieloznaczne (np. słowo „zamek") lub nieunikalne (np. imiona ludzkie) etykiety. Dodatkowo, wyniki *leksykalne* mogą się różnić w zależności od języka, ze względu na różne zestawy homonimów i wiele naturalnych różnic między językami. Mimo to wiele metod odwzorowywania ontologii używa podobieństwa *leksykalnego* w pierwszych krokach odnajdywania połączeń między ontologiami, które nie mają wcześniej zdefiniowanych łączy między nimi. Takie metody odwzorowywania ontologii są czasem nazywane „terminologicznymi" [18].

Nieformalnie, wymiar *leksykalny* określa podobieństwa nazw bytów w słowniku. Niestety, tego typu podobieństwo cierpi z powodu problemu powszechnego w słownikach, czyli niejednoznaczności. Tak zwane „ujednoznacznienie słów" (word sense disambiguation) to duży problem w przetwarzaniu tekstu [30] i ontologiach (np. w zastosowaniu do bytów nazwanych [5]). Niejednoznaczność języka negatywnie wpływa na dokładność oceny podobieństwa leksykalnego. Zauważmy tutaj, że w przypadku dobrze zdefiniowane ontologii, nie ma problemu niejednoznaczności, ponieważ opisy by-

tów są porównywalne bezpośrednio. Porównując terminy słownikowe musimy najpierw dowiedzieć się, jaki podmiot reprezentuje każdy termin (co jest bytem podstawowym), a następnie porównać same byty. Niestety, obydwa standardowe zbiory referencyjne (Miller [29], i Rubenstein [39]) (często używane do oceny metod WordNet'owych) nie zawierają opisów pojęć, a tylko pary słów, bez dodatkowej informacji o znaczeniu, co sprawia, że ocena dokładności miar leksykalnych jest problematyczna.

3.4 Inne wymiary

Wymiary przedstawione w poprzedzających sekcjach są bardzo ogólne i tym samym szeroko stosowalne, ale istnieje wiele innych sposobów podziału wiedzy na wymiary. Każda ze wspomnianych prac [15, 35, 24, 1, 6, 32] używa różnego rodzaju relacji semantycznych i aksjomatów, koncepcyjnie zbliżonych do wymiarów „dostosowanych" do konkretnego problemu. Można nawet stwierdzić, że każda partycja wiedzy stanowi zbiór semantycznych wymiarów. Wymiary zaproponowane powyżej zostały zaprojektowane (na podstawie analizy istniejących metod i ontologii), i są stosunkowo proste w interpretacji i wystarczająco ogólne, aby były dostępne w prawie każdej bazie wiedzy. Istnieją jednak inne, bardziej szczegółowe wymiary, które warto wspomnieć.

Zacznijmy od wymiaru *członkostwa*. Może on być używany do pomiaru podobieństwa (tylko) między klasami przez zbieranie i porównywanie zestawów instancji, które są określonego typu. W porównaniu do innych wymiarów, ten produkuje dane stosunkowo proste, ponieważ przynależność jest predykatem binarnym—osoba jest lub nie jest danego typu. Taka wiedza może być łatwo wykorzystana do konstruowania metod cechowych. Wymiar *członkostwa* jest (niejawnie) używany w [11], gdzie autorzy budują zestawy funkcji składających się z członków i obliczają podobieństwo w sposób bardzo podobny do metody Tversky'ego. Wymiar *członkostwa* używa danych, które częściowo pokrywają się z wymiarem *taksonomicznym*, ale nadal oferuje własną, unikalną perspektywę na podobieństwo.

Osobno, wymiar *opisowy* zawiera wiedzę na temat wszystkich właściwości, bez wyłączeń. Prostym sposobem utworzenia nowego wymiaru podobieństwa jest wyizolowanie zbioru właściwości, które są obecne w wymiarze *opisowym*. Wynikowy zestaw powinien mieć własną interpretację, określoną jako odrębny wymiar. Ponieważ taki wymiar jest zbudowany z wiedzy *opisowej*, można go nazwać *podwymiarem* opisowym.

Przykładem stworzonym w taki właśnie sposób, jest wymiar *kompozycyjny*. Składa się z właściwości, które oznaczają „jest częścią", „posiada części", „posiada składniki" itp. Ma bardzo jasną interpretację i w miarę łatwo jest nieformalnie ocenić „kompozycję", czy też skład obiektów fizycznych. Formalnie ten wymiar jest reprezentowany przez właściwości, takie jak *hasPart*, *isPartOf*, *isIngredient*, itp. W ontologii SSN[1] tego rodzaju relacje są reprezentowane przez właściwość *hasPart* (odziedziczoną z ontologii *DUL*). Podobna właściwość istnieje też w WordNet (również o nazwie *hasPart*), oraz w wielu ontologiach.

Innym przykładem „pod–opisowym" jest wymiar *fizyczny*. Zawiera on wszystkie role opisujące każdy rodzaj fizycznej charakterystyki, właściwości bądź atrybutu. To, jakie konkretne właściwości są zawarte w tym wymiarze zależy od ontologii. Mogą one obejmować wielkość (np. wysokość, szerokość, obszar), masę, kolor, kształt i inne.

Praktycznym problemem z podziałem wymiaru *opisowego* na podwymiary jest to, że zastoso-

[1] `https://www.w3.org/TR/vocab-ssn/`

wanie wymiaru zbudowanego przez tą metodę wymaga określonych właściwości. Nawet kierując się interpretacją, konkretne wymiary mogą być reprezentowane przez różne właściwości w różnych bazach wiedzy. W jednej ontologii wymiar *fizyczny* będzie zawierać właściwości *hasWeight* i *hasHeight*, podczas gdy w innej np. tylko *hasArea*. Jeszcze inna ontologia może nie zawierać żadnych ról istotnych dla wymiaru *fizycznego*, a zatem wynik podobieństwa *fizycznego* nie byłby możliwy do obliczenia. Każdy podział wymiaru *opisowego* zazwyczaj oznacza utratę uniwersalności, to znaczy nie można zastosować naszego nowego wymiaru do każdej ontologii. Kolejną wadą tej metody jest to, że każdy „pod–opisowy" wymiar wiedzy w sposób bardzo oczywisty nakłada się na wymiar *opisowy*. W konsekwencji, na przykład, wynik *kompozycyjny* i *opisowy* nie są niezależne (jeden jest zawarty w drugim), a wymiary nie są ortogonalne. Z drugiej strony, podwymiary opisowe są łatwe do użycia dla metod grafowych, które działają na ontologii, takich jak [13]. W wypadku liczenia odległości ścieżkowej w grafie, wystarczy, że użyjemy tylko krawędzi określonego typu, zamiast wszystkich krawędzi. Trzeba pamiętać, że nie wszystkie typy krawędzi są często używane, tworząc interesujący i użyteczny wymiar.

Podsumowując, podwymiary opisowe, nawet te z jasną interpretacją, jak na przykład wymiar *fizyczny*, charakteryzują się utratą ogólności w porównaniu do wymiaru opisowego. Mogą one być wdrażane inaczej w różnych ontologii, a różnice mogą być na tyle istotne, aby sugerować użycie wymiaru o większej ziarnistości (ang. „granularity"). Ziarnistość wymiarów omówiono bardziej szczegółowo w [45].

W rozprawie został przedstawiony pełny opis podejścia wymiarowego, uzupełniony o rozbudowany przykład obrazujący stosowanie wymiarów podobieństwa w praktyce. Zostały opisane również rozważania na temat łączenia wyników z poszczególnych wymiarów, lub ich bezpośredniej interpretacji, jako pełnoprawne wyniki świadczące o konkretnym sposobie, w jaki byty są podobne.

4 Algorytm podobieństwa

Centralną częścią platformy *SimDim* jest generyczny algorytm podobieństwa, za pomocą którego podejście wymiarowe może zostać wprowadzone w praktykę. Został on zaprojektowany, jako rozszerzenie istniejących podejść o szersze (i wielowymiarowe) spojrzenie na problem podobieństwa. Algorytm odnosi się do problemów zaobserwowanych pośród istniejących rozwiązań (opisanych w rozdziale 1.1) poprzez następujące właściwości:

- **Wielowymiarowość** Nowoczesne źródła danych, w tym ontologie, są znacznie bogatsze w złożone struktury i aksjomaty, niż klasyczne, które czasami składają się tylko z taksonomii lub prostych opisów. Semantycznie adnotowane zasoby również zyskują na popularności i mogą być postrzegane jako instancje ontologiczne. Dobrze ugruntowane standardy adnotacji polegają na pełnych możliwościach ontologii. Klasyczne miary podobieństwa były często budowane dla istniejących wówczas modeli danych, i nie biorą pod uwagę wszystkich informacji zawartych w ontologii. Prowadzi to do sytuacji, w których zmiana w ontologii (np. dodanie setek asercji) nie zmienia wyniku podobieństwa, tylko dlatego, że używana miara działa w wymiarze, który ich nie obejmuje. Jest to znane jako problem *nieczułości na zmiany*. *SimDim* oferuje głęboką parametryzację, za pomocą której dowolna część dostępnych danych może być wzięta pod uwagę podczas obliczania podobieństwa. Dzięki temu *SimDim* umożliwia podejście wymiarowe, w którym rodzaj użytych danych jest podany explicite. Wyniary podobieństwa, które wpływają na wynik mogą być dowolnie dobrane, a ogólna natura

SimDim umożliwia nie tylko użycie wszystkich dostępnych danych, ale także pracę w dowolnej domenie, i z dowolnym kształtem danych. Innymi słowy, *SimDim* nie jest ograniczony do podobieństwa ontologicznego, lecz jest w stanie zrealizować obliczenia w dowolnym zakresie, pod warunkiem, że funkcje będące parametrami mogą być w nim zdefiniowane.

- **Głęboka perspektywa** Chociaż nie jest to prawdą dla wszystkich metod (zwłaszcza tych wyspecjalizowanych dla pojedynczej ontologii), nowoczesne metody oparte o modele cechowe często przejmują atomowy widok na cechy obiektów. Oznacza to, że semantyka cech nie jest brana pod uwagę, a cechy są redukowane do ich atomowej reprezentacji symbolicznej, co jest przejawem *płytkości*. *SimDim* proponuje odmienne podejście, gdzie cecha może mieć swoją własną semantykę i powinna być traktowana jako oddzielny *byt*, który może mieć swoje własne cechy. Idea ta jest realizowana w algorytmie poprzez rekurencję, w której każda cecha może mieć swoje własne pod–cechy, co pozwala na *głębsze* porównanie bytów. *SimDim* jest konfigurowany m.i. za pomocą funkcji cechujących, w celu wsparcia wielu różnych sposobów wyodrębniania cech z obiektów, co umożliwia głęboką analizę bytów.

Algorytm jest parametryzowany za pomocą funkcji cechujących, funkcji podobieństwa, oraz reduktorów. Parametry algorytmu są opisane w poniższych sekcjach.

4.1 Cechowania

Rozpoznawanie cech jest po prostu aktem identyfikacji/ekstrakcji cech z obiektu. W kontekście niniejszej pracy powinno być rozumiane w kategoriach określonych przez cechowe modele podobieństwa. Pomimo faktu, że istnieje wiele różnych sposobów rozpoznawania cech stosowanych w praktyce, pojęcie „funkcji cechującej" nie było wcześniej opisany szczegółowo, lub analizowane. Dobrym przykładem tego problemu jest tzw. Maedche Formula [25] przytoczona w równaniu 2.12, która jawnie używa przodków klasy jako jej zbioru cech. Takie cechowanie można również znaleźć w innych metodach ([36, 43, 38, 4]), lecz mimo tego, cechowanie, ani funkcje cechujące nie są rozpatrywane oddzielnie.

W tej sekcji przedstawiono ideę cechowania, w rozumieniu platformy *SimDim*, oraz przykłady funkcji cechujących.

Definicja 4.1. (Cechowanie) Niech $\mathcal{O}$ będzie pewnym uniwersum obiektów semantycznych, a $\mathbf{FinSet}(\mathcal{O})$ zbiorem wszystkich skończonych podzbiorów $\mathcal{O}$. Funkcja częściowa $f : \mathcal{O} \rightharpoonup \mathbf{FinSet}(\mathcal{O})$, jest zwana **cechowaniem**, lub funkcją cechującą.

Definicja 4.2. (Zbiór cech) Niech $\mathfrak{o}$ będzie dowolnym obiektem semantycznym. $f(\mathfrak{o})$ jest zwany **zbiorem cech** $\mathfrak{o}$ względem cechowania f. Każdy element $x \in f(\mathfrak{o})$, x jest zwany **cechą** $\mathfrak{o}$ (również względem cechowania f).

Mówiąc prościej, cechowanie wyodrębnia skończony (być może pusty) zbiór cech obiektu. Jeśli dla danego argumentu $\mathfrak{o}$ i cechowania f, $f(\mathfrak{o}) = \emptyset$, lub f nie jest zdefiniowane dla $\mathfrak{o}$, mówimy, że f nie rozpoznaje cech w $\mathfrak{o}$, lub, że f nie jest *znacząca* dla $\mathfrak{o}$.

Przykład cechowania dla wyrażeń klasowych jest zdefiniowany w następujący sposób:

Przykład 4.3. (Cechowanie $f_{\sqcap}$) Niech ce będzie wyrażeniem klasowym, a I jakimś zbiorem indeksów.

$$f_{\sqcap}(ce) = \begin{cases} \{ ce_i \mid i \in I \} & \text{if } ce = \bigsqcap_{i \in I} ce_i \\ \{ce\} & \text{w innym wypadku} \end{cases} \tag{4.1}$$

Innymi słowy, jeśli ce ma postać $ce_1 \sqcap \ldots \sqcap ce_n$, to $\{ce_1, \ldots, ce_n\}$ jest zbiorem cech ce względem $f_{\sqcap}$.

Analogicznie może zostać zdefiniowana funkcja $f_{\sqcup}$. Te dwie funkcje dzielą wyrażenia klasowe względem do $\sqcap$, lub $\sqcup$ a każdy element wynikający z takiego podziału jest traktowany jak nowa cecha. Należy zauważyć, że $f_{\sqcap}$ and $f_{\sqcup}$ nie są zdefiniowane dla obiektów innych niż wyrażenia klasowe.

Funkcje cechujące mogą być wymiarowe. Na przykład funkcja, która dla klasy zwraca zestaw jego przodków jest ściśle *taksonomiczna*, ponieważ używa wyłącznie taksonomii do tworzenia cech. Modyfikacja takiej funkcji (która nie jest bezpieczna względem rekurencji) może również zawierać oryginalną klasę w danych wyjściowych. Inną funkcję, która dla klasy lub wyrażenia klasowego zwraca wszystkie ich instancje, należy do wymiaru *członkostwa*.

Przykładem *opisowej* funkcji cechującej jest $f_{\sqsubseteq def}$, zdefiniowana poniżej:

Przykład 4.4. (Cechowanie $f_{\sqsubseteq \text{def}}$) Niech c będzie klasą nazwaną, a NC zbiorem wszystkich klas nazwanych.

$$f_{\sqsubseteq \text{def}}(c) = \{ ce \mid c \sqsubseteq ce, ce \notin NC \} \tag{4.2}$$

Według $f_{\sqsubseteq \text{def}}$ cechy to wyrażenia będące podklasami c, i nie będące klasami nazwanymi (nie należące do NC). To oznacza, że tylko wyliczenia, zbiorowe wyrażenia klasowe (koniunkcje, dysjunkcje i dopełnienia), oraz restrykcje, są cechami (według tej funkcji). Tym sposobem $f_{\sqsubseteq def}$ tworzy cechy będące częścią definicji c, ale nie będące bezpośrednią częścią taksonomii.

Wbrew temu, co mogą sugerować powyższe przykłady, funkcje cechowania niekoniecznie muszą być wymiarowe. Na przykład $f_{\sqcap}$ i $f_{\sqcup}$ zdefiniowane wcześniej tworzą w wyniku wyrażenia klasowe, które nie są przypisane wyłączne do jednego wymiaru podobieństwa. W związku z tym możemy powiedzieć, że te funkcje są agnostyczne względem wymiarów i mogą być używane w dowolnym wymiarze.

4.2 Funkcje podobieństwa

Kolejnym parametrem algorytmu jest zbiór funkcji podobieństwa. Mimo tego, że wiele funkcji podobieństwa jest jednocześnie miarami podobieństwa, te pojęcia są oddzielne. Funkcje podobieństwa nie muszą wykazywać właściwości, które są wymagane od miar podobieństwa zdefiniowanych niezależnie, w innych pracach. Funkcje podobieństwa, rozumiane w kontekście *SimDim*, są zdefiniowane następująco:

Definicja 4.5. (Funkcja podobieństwa) **Funkcja podobieństwa**n $s : \mathcal{O} \times \mathcal{O} \rightharpoonup [0, 1]$, gdzie $\mathcal{O}$ to dane uniwersum obiektów semantycznych, to funkcja, która wykazuje następującą właściwość:

- **tożsamość** $\forall \mathfrak{o}_1, \mathfrak{o}_2 \; (s(\mathfrak{o}_1, \mathfrak{o}_2) = 1 \Leftrightarrow \mathfrak{o}_1 = \mathfrak{o}_2)$

Dlatego, że warunki stawiane funkcji s są mało restrykcyjne, wiele funkcji przedstawionych w sekcji 2 je spełnia. Jedną z najprostszych funkcji podobieństwa jest $s_{=}$, nazywana „komparatorem binarnym":

Przykład 4.6. (Funkcja podobieństwa $s_=$) Niech $\mathfrak{o}_1$ i $\mathfrak{o}_2$ będą dowolnymi bytami.

$$s_=(\mathfrak{o}_1, \mathfrak{o}_2) = \begin{cases} 1 & \text{if } \mathfrak{o}_1 = \mathfrak{o}_2 \\ 0 & \text{w innym wypadku} \end{cases} \tag{4.3}$$

Podobnie, jak cechowania, funkcje podobieństwa mogą być wymiarowe i być zdefiniowane tylko dla konkretnego rodzaju typów. Przykładem taksonomicznej funkcji cechowania, jest funkcja oparta o definicję Lin'a (równanie 2.8), zdefiniowana w następujący sposób:

$$s_{LinWN}(\mathfrak{o}_1, \mathfrak{o}_2) = \frac{2 \cdot IC(LCS(\mathfrak{o}_1, \mathfrak{o}_2))}{IC(\mathfrak{o}_1) + IC(\mathfrak{o}_2)} \tag{4.4}$$

W powyższym równaniu $\mathfrak{o}_1$ i $\mathfrak{o}_2$ to klasy ontologiczne, więc s_{LinWN} jest zdefiniowana tylko dla bytów tego rodzaju. Pojęcia LCS i IC zostały opisane przy równaniu 2.8.

4.3 Reduktory

Reduktory to funkcje używane przez algorytm do sprowadzania poszczególnych wyników podobieństwa do jednej wartości. Reduktory są zdefiniowane w następujący sposób:

Definicja 4.7. (Reduktor) **Reduktor** to funkcja $\mathtt{r} : \mathbf{FinMulSet}([0,1]) \to [0,1]$, gdzie $\mathbf{FinMulSet}([0,1])$ to zbiór wszystkich skończonych pod-wielozbiorów $[0,1]$. Reduktory muszą wykazywać następujące właściwości:

- **zachowanie zbioru jednoelementowego** $\forall x \in [0,1]\ (\mathtt{r}(\{x\}) = x)$
- **zachowanie wartości pustej** $\mathtt{r}(\emptyset) = 0$

Reduktory mogą być konstruowane z prostych funkcji matematycznych. Dla przykładu:

Przykład 4.8. (reduktor $\mathtt{r}_{\max}$) $\mathtt{r}_{\max}(M) = \max(M)$

Powyższe równanie prezentuje reduktor, który wybiera największą wartość z wielozbioru M.

Inne proste reduktory można stworzyć używając różnorakich statystyk, np. mediany, średniej, itd. Nieco bardziej skomplikowane reduktory traktują wartości zerowe w specjalny sposób, nie biorąc ich pod uwagę przy obliczaniu wyniku.

4.4 Algorytm

Ogólnie rzecz biorąc, algorytm stanowiący centralną część platformy *SimDim* proponuje podejście rekurencyjne, gdzie obiekty są najpierw rekurencyjnie cechowane, a każda cecha jest przedstawiana jako nowy obiekt wejściowy, w kolejnych krokach rekurencji. Gdy dla danego cechowania nie są rozpoznawane nowe cechy, należy użyć następnej funkcji cechującej (ze zbioru przygotowanego wcześniej, według kryteriów określonych w algorytmie). Ostatecznie, kiedy wszystkie cechowania zostaną „wyczerpane" pewna funkcja podobieństwa (wybierana według algorytmu) jest używana aby obliczyć *głębokie* podobieństwo cech. Podczas „zwijania" (cofania) stosu rekurencji pośrednie wyniki podobieństwa są zbierane przy użyciu reduktora i podsumowane za pomocą funkcji 4.6.

Zdefiniowanie algorytmu użytego w *SimDim* wymaga wprowadzenia szeregu definicji pomocniczych. Pierwszą z nich jest podobieństwo zbiorów cech.

Załóżmy, że zdefiniowano jakieś cechowania f i g, $\mathfrak{o}_1$ and $\mathfrak{o}_2$ to obiekty semantyczne, a $f(\mathfrak{o}_1) = \mathfrak{O}_1$ and $g(\mathfrak{o}_2) = \mathfrak{O}_2$ to zbiory cech.

Definicja 4.9. (Podobieństwo $\mathfrak{O}_1$ do $\mathfrak{O}_2$ względem s i $\mathtt{r}$) $\mathrm{SimSet}^{\mathtt{r}}_s(\mathfrak{O}_1, \mathfrak{O}_2)$ to podobieństwo zbioru cech $\mathfrak{O}_1$ do zbioru cech $\mathfrak{O}_2$, przy użyciu funkcji podobieństwa s i reduktora $\mathtt{r}$.

$$\mathrm{SimSet}^{\mathtt{r}}_s(\mathfrak{O}_1, \mathfrak{O}_2) = \{\, 1/x \mid x \in \mathfrak{O}_1 \,\} \cup \{\, \mathtt{r}(s(\_, y))/y \mid y \in \mathfrak{O}_2 \,\} \tag{4.5}$$

Gdzie $s(\_, y) = \{\, s(x, y) \mid x \in \mathfrak{O}_1 \wedge (x, y) \in dom(s) \,\}$, a $\{m/x\}$ oznacza zbiór rozmyty, w którym element x ma wartość funkcji przynależności równą m $(0 \leq m \leq 1)$.

$\mathrm{SimSet}^{\mathtt{r}}_s(\mathfrak{O}_1, \mathfrak{O}_2)$ to zbiór rozmyty zawierający wszystkie cechy $\mathfrak{O}_1$, oraz te cechy z $\mathfrak{O}_2$, które są podobne (według funkcji s) do przynajmniej jednej cechy $\mathfrak{O}_1$. W $\mathrm{SimSet}^{\mathtt{r}}_s(\mathfrak{O}_1, \mathfrak{O}_2)$ wartość przynależności dla cech $\mathfrak{O}_1$ zawsze wynosi 1, a dla cech $\mathfrak{O}_2$ zależy od tego, jak podobne są do cech $\mathfrak{O}_1$. Dla każdej cechy z $\mathfrak{O}_2$ podobieństwo do wszystkich cech $\mathfrak{O}_1$ jest obliczane parami za pomocą funkcji s, a reduktor $\mathtt{r}$ jest używany na powstałym wielozbiorze wyników, aby określić ostateczną wartość funkcji przynależności. Funkcja $\mathrm{SimSet}^{\mathtt{r}}_s(\mathfrak{O}_1, \mathfrak{O}_2)$ nie jest symetryczna.

Względne podobieństwo zbiorów podobieństwa cech jest użyte w następującej funkcji, inspirowanej indeksem Jaccarda:

$$\mathit{SimDim}^{\mathtt{r}}_s(\mathfrak{O}_1, \mathfrak{O}_2) = \frac{|\mathrm{SimSet}^{\mathtt{r}}_s(\mathfrak{O}_1, \mathfrak{O}_2) \cap \mathrm{SimSet}^{\mathtt{r}}_s(\mathfrak{O}_2, \mathfrak{O}_1)|}{|\mathfrak{O}_1 \cup \mathfrak{O}_2|} \tag{4.6}$$

Zależnie od doboru parametrów, algorytm ma różne właściwości i potrafi dogłębnie zbadać podobieństwo bytów różnego rodzaju. *Instancją SimDim* nazywamy instancję algorytmu, oznaczaną jako $\mathcal{G}(\mathtt{r}, \mathrm{sf}, \mathrm{ff})$, gdzie $\mathtt{r}$ to określony reduktor, a sf i ff to określone, skończone zbiory odpowiednio funkcji podobieństwa i cechowań. Zależnie od parametrów, instancje *SimDim* mogą działać w jednym lub wielu wymiarach podobieństwa.

W celu wybrania, które funkcje podobieństwa i cechowania powinny zostać zastosowane w każdym kroku, algorytm stosuje następujące definicje:

Definicja 4.10. (Najodpowiedniejsza funkcja podobieństwa) Niech $\mathfrak{o}_1, \mathfrak{o}_2$ będą obiektami semantycznymi, a ssf będzie skończonym podzbiorem funkcji podobieństwa. Funkcja podobieństwa *najodpowiedniejsza* dla ssf, $\mathfrak{o}_1$ i $\mathfrak{o}_2$ to funkcja $s \in \mathrm{ssf}$ zdefiniowana dla $(\mathfrak{o}_1, \mathfrak{o}_2)$, która wykazuje następującą właściwość:

- **Minimalizacja typu** $\neg\exists_{t \in \mathrm{ssf}}\, \mathrm{dom}(t) \subset \mathrm{dom}(s)$

Oznacza to, że taka funkcja musi być jak najbardziej specyficzna spośród sff dla jej parametrów $\mathfrak{o}_1$ i $\mathfrak{o}_2$.

Ta definicja implikuje istnienie funkcji bestSim : $\mathbf{FinSet}(\mathtt{SF}) \to \mathcal{O} \times \mathcal{O} \to \mathbf{FinSet}(\mathtt{SF})$, gdzie bestSim(ssf)$(\mathfrak{o}_1, \mathfrak{o}_2)$ to zbiór funkcji podobieństwa najodpowiedniejszych dla ssf, $\mathfrak{o}_1$ i $\mathfrak{o}_2$.

Analogiczna definicja została stworzona dla cechowań:

Definicja 4.11. (Najodpowiedniejsze cechowanie) Niech $\mathfrak{o}$ będzie obiektem semantycznym, a sff skończonym zbiorem cechowań. Cechowanie *najodpowiedniejsze* dla sff i $\mathfrak{o}$ to funkcja $f \in \mathrm{sff}$ wykazująca następującą właściwość:

- **Minimalizacja typu** $\neg\exists_{g \in \mathrm{sff}}\, \mathrm{dom}(g) \subset \mathrm{dom}(f)$

Funkcja pomocnicza bestFeat : $\mathbf{FinSet}(\mathtt{FF}) \to \mathcal{O} \to \mathbf{FinSet}(\mathtt{FF})$ definiuje bestFeat(sff)$(\mathfrak{o})$ jako zbiór cechowań z sff najodpowiedniejszych dla $\mathfrak{o}$.

Prezentacja algorytmu wymaga zdefiniowania trzech kolejnych funkcji, które wykorzystują pojęcie najodpowiedniejszych funkcji:

Definicja 4.12. (blendSim) Niech $\mathfrak{o}_1, \mathfrak{o}_2$ będą obiektami semantycznymi, sf skończonym zbiorem funkcji podobieństwa, a $\mathtt{r}$ reduktorem.

$$\text{blendSim}^{\mathtt{r}}_{\text{sf}}(\mathfrak{o}_1, \mathfrak{o}_2) = \begin{cases} \mathtt{r}(\{\, s(\mathfrak{o}_1, \mathfrak{o}_2) \mid s \in \text{bs} \,\}) & \text{if bs} \neq \emptyset \\ s_{=}(\mathfrak{o}_1, \mathfrak{o}_2) & \text{otherwise} \end{cases} \tag{4.7}$$

gdzie bs = bestSim(sf)$(\mathfrak{o}_1, \mathfrak{o}_2)$.

blendSim to, upraszczając, reduktor zastosowany do wyników najodpowiedniejszych funkcji dla sf, $\mathfrak{o}_1$ i $\mathfrak{o}_2$. W specjalnym przypadku, kiedy zbiór najodpowiedniejszych funkcji jest pusty, stosowana jest funkcja $s_{=}$.

Definicja 4.13. (blendSimRec) Niech $\mathfrak{o}_1, \mathfrak{o}_2$ będą obiektami semantycznymi, sf skończonym zbiorem funkcji podobieństwa, a $\mathtt{r}$ reduktorem.

$$\text{blendSimRec}^{\mathtt{r}}_{\text{sf,ff}}(\mathfrak{o}_1, \mathfrak{o}_2) = \begin{cases} \mathtt{r}(\{\, s(\mathfrak{o}_1, \mathfrak{o}_2) \mid s \in \text{bs} \,\}) & \text{if bs} \neq \emptyset \\ \mathcal{G}(\mathtt{r}, \text{sf}, \text{ff})(\mathfrak{o}_1, \mathfrak{o}_2) & \text{otherwise} \end{cases} \tag{4.8}$$

gdzie bs = bestSim(sf)$(\mathfrak{o}_1, \mathfrak{o}_2)$.

blendSimRec to wersja blendSim, w której instancja *SimDim* jest użyta zamiast $s_{=}$. W związku z tym, że instancja *SimDim* jest jednoznacznie identyfikowana przez jej parametry, a $\mathcal{G}(\mathtt{r}, \text{sf}, \text{ff})$ dziedziczy parametry po $\text{blendSimRec}^{\mathtt{r}}_{\text{sf,ff}}$, oznacza to, że występująca tu instancja *SimDim* jest stosowana rekurencyjnie.

W sposób analogiczny do blendSim, zdefiniowana jest następujące funkcja dla cechowań:

Definicja 4.14. (blendFeat) Niech $\mathfrak{o}_1$ będzie obiektem semantycznym, a ff skończonym zbiorem cechowań.

$$\text{blendFeat}_{\text{ff}}(\mathfrak{o}_1) = \begin{cases} \bigcup\{\, f(\mathfrak{o}_1) \mid f \in \text{bf} \,\} & \text{if bf} \neq \emptyset \\ \emptyset & \text{otherwise} \end{cases} \tag{4.9}$$

gdzie bf = bestFeat(ff)$(\mathfrak{o}_1)$.

blendFeat oblicza sumę zbiorów cech rozpoznanych przez najodpowiedniejsze cechowania. W przypadku specjalnym, gdy najodpowiedniejsze cechowania nie istnieją, wynik jest zbiorem pustym.

Z pomocą definicji opisanych powyżej, algorytm używany w *SimDim* jest zdefiniowany następująco:

Podsumowując, algorytm wykonuje porównanie bytów parami, poprzez, w pierwszej kolejności, obliczenie cech, następnie policzenie podobieństwa zbiorów cech za pomocą równania 4.6. Sposób, w jaki te obliczenia są wykonywane, zależy od doboru parametrów. Jeśli, dla danego rodzaju bytów, wśród parametrów znajdują się pasujące funkcje cechowania, algorytm stosuje cechowanie rekurencyjnie tzn. traktując obliczone w danym kroku cechy, jak pełnoprawne byty i stosując dla nich cechowania. Algorytm obejmuje również przypadki specjalne, w których odpowiednie cechowania lub funkcje podobieństwa nie istnieją.

Algorytm 1 Algorytm użyty w *SimDim*

parametry instancji *SimDim*:
sf $\leftarrow$ zbiór funkcji podobieństwa
ff $\leftarrow$ zbiór cechowań
$\mathtt{r} \leftarrow$ reduktor
argumenty:
$(\mathfrak{o}_1, \mathfrak{o}_2) \leftarrow$ para porównywanych bytów

if $\mathfrak{o}_1 = \mathfrak{o}_2$ **then** ▷ Przypadek specjalny: $\mathfrak{o}_1$ and $\mathfrak{o}_2$ są równe
    return 1

fx $\leftarrow \mathrm{blendFeat}_{\mathrm{ff}}(\mathfrak{o}_1)$
fy $\leftarrow \mathrm{blendFeat}_{\mathrm{ff}}(\mathfrak{o}_2)$
if fx = fy = $\emptyset$ **then** ▷ Przypadek specjalny: cechy $\mathfrak{o}_1$ lub $\mathfrak{o}_2$ nie są rozpoznane
    return $\mathrm{blendSim}^{\mathtt{r}}_{\mathrm{sf}}(\mathfrak{o}_1, \mathfrak{o}_2)$

bsr $\leftarrow \mathrm{blendSimRec}^{\mathtt{r}}_{\mathrm{sf,ff}}$
return $\mathit{SimDim}^{\mathtt{r}}_{\mathrm{bsr}}(\mathrm{fx}, \mathrm{fy})$

Rekursywne użycie *SimDim* jest możliwe, ze względu na fakt, że każda konkretna instancja *SimDim* jest funkcją podobieństwa, więc może zostać zawsze użyta tam, gdzie można użyć takiej właśnie funkcji.

Głęboka parametryzacja algorytmu oznacza, że mogą być stworzone bardzo różne instancje *SimDim*. Dla przypadku, w którym zbiory cechowań i funkcji podobieństwa będące parametrami instancji, są puste, instancja $\mathcal{I}_{\mathrm{empty}} = \mathcal{G}(\mathtt{r}_{\max}, \emptyset, \emptyset)$ jest równoważna z funkcją $s_{=}$. Wiele ciekawszych instancji dla różnego rodzaju bytów może zostać zdefiniowanych przez umiejętny dobór parametrów.

Dla przykładu, załóżmy, że istnieje funkcja podobieństwa obiektów s_{label}, która porównuje etykiety bytów używając WordNet'u, za pomocą jednej z WordNet'owych miar zdefiniowanych w literaturze. Instancja *SimDim* dla wymiaru leksykalnego, która porównuje klasy, wiązałaby się ze stworzeniem cechowania f_{label}, która dla dowolnej klasy ontologicznej zwraca zbiór jej etykiet, pobieranych np. z wartości konkretnych adnotacji (e.g. rdfs:label), lub części identyfikatora URI. Instancja $\mathcal{G}(\mathtt{r}_{\max}, \{s_{\mathrm{label}}\}, \{f_{\mathrm{label}}\})$ (należąca do wymiaru leksykalnego) i realizuje cechowanie klas do etykiet i ich porównanie za pomocą WordNet'u. Do tak zdefiniowanej instancji można oczywiście dokładać kolejne cechowania, lub nawet funkcje podobieństwa, zwiększając zakres jej działania i obsługiwane rodzaje bytów.

Pełna rozprawa zawiera opis wielu przykładowych cechowań, funkcji podobieństwa, reduktorów, oraz instancji *SimDim*. Zawarty został również poszerzony opis działania algorytmu oraz wpływu parametrów na jego działanie.

5 Podsumowanie

Rozprawa zawiera krytyczną analizę obecnego stanu wiedzy (opisaną w rozdziale 2.), prezentującą meta-poziomowe spojrzenie na istniejące modele i miary podobieństwa. Analiza służy jako podsumowanie istniejących podejść, ich założeń, ograniczeń i problemów. Wiedza zdobyta podczas badań nad stanem wiedzy posłużyła do stworzenia wymiarowego podejścia do obliczania podobieństwa, oraz do zaprojektowania platformy platformy *SimDim*, jako autorskiego model obliczania podobieństwa, zawierającego generyczny algorytm (wraz z implementacją). Podejście wymiarowe zostało szczegółowo opisane w rozdziale 3. rozprawy. Ogólna definicja i opis platformy *SimDim*, wraz z definicją generycznego algorytmu została przedstawiona w rozdziale 4. Szczegóły dotyczące implementacji algorytmu zostały zawarte w rozdziale 5., który wspierany jest przez bogaty zestaw przykładowych parametrów – cechowań, funkcji podobieństwa, oraz reduktorów opisanych w załączniku A.

Wymiary podobieństwa to sposób kategoryzacji metod, modeli, algorytmów i miar podobieństwa, który jest szeroko stosowalny – dla porównania par i grup bytów ontologicznych, dokumentów i terminów, odwzorowywania ontologii, oraz w wielu innych obszarach zastosowania miar podobieństwa. Pozwala określić nie tylko *jak bardzo* podobne są obiekty semantyczne, ale także *w jaki sposób*. Wymiarowe opisanie wyniku zmniejsza wrodzoną niejednoznaczność w interpretacji podobieństwa. Podobieństwo wymiarowe pozwala na bardziej znaczące porównanie wyników między różnymi miarami i umożliwia nowe spojrzenie na wyniki podobieństwa.

Każdy wymiar podobieństwa ma interpretację, która może być uszczegółowiona w zależności od kontekstu i jest niezależna od formatu danych używanego do przechowywania wiedzy. Tak długo, jak ta interpretacja jest przestrzegana, wiele różnych miar może służyć do reprezentowania każdego wymiaru. Ponadto, właściwe wykorzystanie i interpretacja konkretnego wymiaru zależy od intuicyjnego zrozumienia ogólnego opisu tego wymiaru. W ten sposób podejście wymiarowe odzwierciedla subiektywny charakter podobieństwa. Szczegóły podejścia wymiarowego, wraz rozbudowanym przykładem podobieństwa wymiarowego zostały opisane w rozdziale 3. rozprawy.

Algorytm zastosowany w *SimDim* (opisany w rozdziale 4. rozprawy) opiera się na wymiarach teorii podobieństwa, proponując generyczny sposób obliczania podobieństwa w ontologiach, lub dowolnej innej domenie, w której istnieją, lub mogą zostać stworzone reduktory, cechowania i funkcje podobieństwa. Wysoce konfigurowalny, przyjmuje jako parametry funkcje, które cechują obiekty, porównują cechy i podsumowują cząstkowe wyniki podobieństwa cech. Dobór parametrów algorytmu zmienia jego właściwości i pozwala na odzwierciedlenie podejścia wymiarowego, poprzez tworzenie instancji pracujących w konkretnych wymiarach podobieństwa. Częścią pracy jest implementacja algorytmu, oraz kolekcja przykładowych instancji *SimDim*, oraz parametrów, z których można budować nowe instancje. Szczegóły implementacji oraz wytłumaczenie decyzji projektowych dotyczących algorytmu są opisane w rozdziale 5., a przykładowe dostępne funkcje parametryzujące algorytm, oraz konkretne instancje *SimDim* w załączniku A rozprawy.

Podsumowując, główny wkład pracy to zaproponowane wymiarowe podejście do obliczania podobieństwa oraz platforma *SimDim* wraz z implementacją generycznego algorytmu i przykładami dla języka OWL. Zbiorowo, uzyskane wyniki w pełni realizują postawione cele badawcze.

Literatura

[1] E. Albacete, J. Calle, E. Castro, and D. Cuadra. Semantic similarity measures applied to an ontology for human-like interaction. *Journal of Artificial Intelligence Research*, 44:397–421, 2012. 10, 11

[2] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25, 2000. 1, 8

[3] D. Bollegala, Y. Matsuo, and M. Ishizuka. A relational model of semantic similarity between words using automatically extracted lexical pattern clusters from the web. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 803–812. Association for Computational Linguistics, 2009. 4

[4] H. Bulskov, R. Knappe, and T. Andreasen. On measuring similarity for conceptual querying. In J. G. Carbonell, J. Siekmann, T. Andreasen, H. Christiansen, A. Motro, and H. Legind Larsen, editors, *Flexible Query Answering Systems*, pages 100–111, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. 13

[5] R. Bunescu and M. Paşca. Using encyclopedic knowledge for named entity disambiguation. In *11th conference of the European Chapter of the Association for Computational Linguistics*, 2006. 10

[6] F. Calle, E. Castro, and D. Cuadra. Ontological dimensions applied to natural interaction. In *Ontologies in Interactive Systems, 2008. ONTORACT'08. First International Workshop on*, pages 91–96. IEEE, 2008. 11

[7] V. Cross. Fuzzy semantic distance measures between ontological concepts. In *Fuzzy Information, 2004. Processing NAFIPS'04. IEEE Annual Meeting of the*, volume 2, pages 635–640. IEEE, 2004. 7

[8] V. Cross. Tversky's parameterized similarity ratio model: A basis for semantic relatedness. In *Fuzzy Information Processing Society, 2006. NAFIPS 2006. Annual meeting of the North American*, pages 541–546. IEEE, 2006. 7

[9] V. Cross and X. Yu. Investigating ontological similarity theoretically with fuzzy set theory, information content, and tversky similarity and empirically with the gene ontology. In *International Conference on Scalable Uncertainty Management*, pages 387–400. Springer, 2011. 7

[10] C. d'Amato, N. Fanizzi, and F. Esposito. A semantic similarity measure for expressive description logics. *arXiv preprint arXiv:0911.5043*, 2009. 6

[11] C. d'Amato, N. Fanizzi, and F. Esposito. A semantic similarity measure for expressive description logics. *arXiv preprint arXiv:0911.5043*, 2009. 11

[12] C. d'Amato, S. Staab, and N. Fanizzi. On the influence of description logics ontologies on conceptual similarity. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 48–63. Springer, 2008. 6

[13] T. De Nies, C. Beecks, F. Godin, W. De Neve, G. Stepien, D. Arndt, L. De Vocht, R. Verborgh, T. Seidl, E. Mannens, et al. A distance-based approach for semantic dissimilarity in knowledge graphs. In *Semantic Computing (ICSC), 2016 IEEE Tenth International Conference on*, pages 254–257. IEEE, 2016. 12

[14] J. Euzenat and P. Shvaiko. *Ontology Matching*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007. 8

[15] J. Euzenat, P. Valtchev, et al. Similarity-based ontology alignment in owl-lite. In *ECAI*, volume 16, page 333, 2004. 8, 11

[16] L. Feiyu. *State of the art: automatic ontology matching*. Tekniska Högskolan, 2007. 8

[17] C. Fellbaum. *WordNet*. Wiley Online Library, 1998. 1

[18] S. Harispe, S. Ranwez, S. Janaqi, and J. Montmain. Semantic similarity from natural language and ontology analysis. *Synthesis Lectures on Human Language Technologies*, 8(1):1–254, 2015. 3, 7, 10

[19] S. Harispe, D. Sánchez, S. Ranwez, S. Janaqi, and J. Montmain. A framework for unifying ontology-based semantic similarity measures: A study in the biomedical domain. *Journal of biomedical informatics*, 48:38–53, 2014. 9

[20] P. Jaccard. *Distribution de la Flore Alpine dans le Bassin des Dranses et dans quelques régions voisines*, volume 37. Societe Vaudoise des Sciences Naturelles, 01 1901. 4

[21] M. Jarmasz and S. Szpakowicz. Roget's thesaurus and semantic similarity. *Recent Advances in Natural Language Processing III: Selected Papers from RANLP*, 2003:111, 2004. 10

[22] Y. R. Jean-Mary, E. P. Shironoshita, and M. R. Kabuka. Ontology matching with semantic verification. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):235–251, 2009. 8

[23] J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*, 1997. 9

[24] D. Lin et al. An information-theoretic definition of similarity. In *Icml*, volume 98, pages 296–304. Citeseer, 1998. 3, 5, 8, 11

[25] A. Maedche, B. Motik, N. Silva, and R. Volz. MAFRA - a mapping framework for distributed ontologies. In *EKAW '02: Proc. of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, pages 235–250, London, UK, 2002. Springer-Verlag. 13

[26] A. Maedche and S. Staab. *Comparing ontologies-similarity measures and a comparison study*. AIFB, 2001. 6

[27] A. G. Maguitman, F. Menczer, H. Roinestad, and A. Vespignani. Algorithmic detection of semantic similarity. In *Proceedings of the 14th international conference on World Wide Web*, pages 107–116. ACM, 2005. 9

[28] G. K. Mazandu and N. J. Mulder. It-gom: an integrative tool for ic-based go semantic similarity measures. Technical report, Technical report, University of Cape Town (South Africa), 2011. 6

[29] G. A. Miller and W. G. Charles. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28, 1991. 11

[30] A. R. Pal and D. Saha. Word sense disambiguation: A survey. *arXiv preprint arXiv:1508.01346*, 2015. 10

[31] V. Pekar and S. Staab. Taxonomy learning: factoring the structure of a taxonomy into a semantic classification decision. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics, 2002. 9

[32] C. Pesquita, D. Faria, A. O. Falcao, P. Lord, and F. M. Couto. Semantic similarity in biomedical ontologies. *PLoS computational biology*, 5(7):e1000443, 2009. 8, 11

[33] G. Pirro and J. Euzenat. A semantic similarity framework exploiting multiple parts-of speech. In *OTM Confederated International Conferences„ On the Move to Meaningful Internet Systems"*, pages 1118–1125. Springer, 2010. 5

[34] R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *IEEE transactions on systems, man, and cybernetics*, 19(1):17–30, 1989. 4

[35] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *the VLDB Journal*, 10(4):334–350, 2001. 8, 11

[36] S. Ranwez, V. Ranwez, J. Villerd, and M. Crampes. Ontological distance measures for information visualisation on conceptual maps. In R. Meersman, Z. Tari, and P. Herrero, editors, *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, pages 1050–1061, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. 13

[37] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*, 1995. 5

[38] M. A. Rodriguez and M. J. Egenhofer. Determining semantic similarity among entity classes from different ontologies. *IEEE Transactions on Knowledge and Data Engineering*, 15(2):442–456, March 2003. 6, 13

[39] H. Rubenstein and J. B. Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, 1965. 11

[40] A. Schlicker, F. S. Domingues, J. Rahnenführer, and T. Lengauer. A new measure for functional similarity of gene products based on gene ontology. *BMC Bioinformatics*, 7(1):302, Jun 2006. 6

[41] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, July 1948. 5

[42] P. Shvaiko and J. Euzenat. Ontology matching: state of the art and future challenges. *IEEE Transactions on knowledge and data engineering*, 25(1):158–176, 2013. 7, 8

[43] D. Sánchez, M. Batet, D. Isern, and A. Valls. Ontology-based semantic similarity: A new feature-based approach. *Expert Systems with Applications*, 39(9):7718 – 7728, 2012. 13

[44] P. Szmeja, M. Ganzha, M. Paprzycki, and W. Pawlowski. Dimensions of ontological similarity. In *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*, pages 246–249, Feb 2016. 7

[45] P. Szmeja, M. Ganzha, M. Paprzycki, and W. Pawłowski. Dimensions of semantic similarity. In A. E. Gaweda, J. Kacprzyk, L. Rutkowski, and G. G. Yen, editors, *Advances in Data Analysis with Computational Intelligence Methods: Dedicated to Professor Jacek Żurada*, pages 87–125. Springer International Publishing, Cham, 2018. 7, 12

[46] A. Tversky. Features of similarity. *Psychological review*, 84(4):327, 1977. 4

[47] S. Wan and R. A. Angryk. Measuring semantic similarity using wordnet-based context vectors. In *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, pages 908–913. IEEE, 2007. 4

[48] H. Wu, Z. Su, F. Mao, V. Olman, and Y. Xu. Prediction of functional modules based on comparative genome analysis and gene ontology application. *Nucleic acids research*, 33(9):2822–2837, 2005. 5