UNIWERSYTET GDAŃSKI
Wydział Matematyki, Fizyki i Informatyki

Wojciech Denejko
nr albumu: 214 300

Rozpoznawanie tekstu w aplikacjach mobilnych

Praca magisterska na kierunku:
INFORMATYKA
Promotor:
dr Tomasz Borzyszkowski

Gdańsk 2017

Streszczenie

Niniejsza praca ma na celu stworzenie aplikacji rozpoznającej tekst w języku polskim oraz angielskim, charakteryzująca się kompatybilnością z systemami iOS oraz Android. Do wytworzenia aplikacji zostanie użyta platforma Xamarin, która służy do tworzenia aplikacji wieloplatformowych. Zbadane zostaną różne metody połączenia technologi wieloplatformowej z istniejącymi rozwiązaniami OCR. Przedstawiona konwolucyjna sieć neuronowa zaprezentuje klasyfikacje polskiego alfabetu.

Integralną częścią pracy będzie aplikacja OCRecognizer, w której zaimplementowano metody klasyfikacji obrazów. Program umożliwia zrobienie zdjęcia, a następnie przy użyciu kilku opcji, rozpoznanie tekstu.

Słowa kluczowe

C#, Xamarin, .NET, Uczenie maszynowe, Sieci neuronowe, kNN,

Spis treści

Wprowadzenie

Xamarin to platforma deweloperska służąca do tworzenia natywnych aplikacji mobilnych dla systemów iOS, Android oraz Windows, za pomocą wpólnej technologii .NET i języka C# . Dzięki temu możliwe jest uzyskanie do stu procent wspólnego kodu między różnymi platformami. Aplikacje napisane przy użyciu technologii Xamarin i C# mają pełny dostęp do interfejsów, API oraz możliwość tworzenia natywnych interfejsów użytkownika.

Ze względu na dynamiczny rozwój rynku IT, uczenie maszynowe staje się coraz bardziej popularne a algorytmy zyskują lepszą skuteczność dzięki dostępności danych oraz szybszych podzespołów komputerowych.

Urządzenia przenośne mają stosunkowo ograniczone zasoby w związku z tym istnieje problem powiązania tych dwóch dziedzin. Algorytmy systemów uczących się wymagają dużej mocy obliczeniowej. Aplikacje wieloplatformowe pozwalają zaoszczędzić czas na implementacji oraz skuteczniej tworzyć funkcjonalności rozpoznawania tekstu. Połączenie tej technologi z algorytmem służącym do klasyfikacji znaków w obrazie jest bardziej optymalne niż ich natywne odpowiedniki.

Celem pracy jest zbadanie istniejących rozwiązań służących do rozpoznawania tekstu oraz stworzenie sieci neuronowej pozwalającej na klasyfikację znaków pisanych charakterystycznych dla współczesnego języka polskiego. Ponieważ pozyskanie danych z polskimi znakami potrzebnych do trenowania sieci neuronowej stanowi problem, zostało stworzone narzędzie do odczytywania znaków z kartki papieru, a następnie zapisanie ich w formie obrazu 32x32 piksele, w skali szarości.

ROZDZIAŁ 1

Rozpoznawanie tekstu w aplikacjach wieloplatformowych

SGML [**?**] jest to *metajęzyk* służący do opisywania struktury i zawartości dokumentów (standard ISO 8879). Do podstawowych zalet takiego podejścia należy:

- jest to międzynarodowy standard dostępny na wielu platformach sprzętowo-systemowych;
- jest to język opisu *każdego* dokumentu, o praktycznie nieograniczonych możliwościach (*rozszerzalność*)
- umożliwia powtórne wykorzystywanie dokumentów, także w sposób inny od poprzedniego (np. tradycyjna książka i dokument multimedialny utworzony z tego samego dokumentu SGML-owego).

Standard służy jedynie do opisywania logicznej struktury dokumentów, nie determinuje ostatecznej formy prezentacji informacji, która może być docelowo przekształcana w najróżniejszy sposób. Dokument zakodowany z wykorzystaniem SGML-a może służyć jako postać wyjściowa do formatowania tej samej informacji w różny sposób i prezentacji z użyciem różnych mediów np. w formie drukowanej na papierze, w postaci hipertekstu albo tekstowej bazy danych. Pozwala to zminimalizować koszty, cały cykl wydawniczy dokonuje się na jednym dokumencie – pliku SGML-owym, a nie na wielu.

1.1. Przedstawienie problemu

Standard SGML to specyfikacja techniczna metajęzyka. Zaś systemem SGML nazywamy *zestaw narzędzi i środków niezbędnych do tworzenia, składowania i obróbki dokumentów z wykorzystaniem tego standardu*. Typowy proces produkcji do-

kumentów w standardzie SGML podzielony jest na kilka części. Najważniejszymi elementami tego procesu są [?, s. 45–47]:

- Klasyfikacja tworzonych dokumentów w grupy i wynikający z tego dobór definicji typu dokumentu[1] (por. punkt 1.2).
- Wybraniu odpowiednich narzędzi do tworzenia i modyfikowania dokumentów SGML (edytory, konwertery, por. punkt 1.4, s. 10)[2].
- Sprawdzenie poprawności oznaczenia dokumentów (walidacja).
- Ustalenie metod składowania zbiorów oznakowanych dokumentów.
- Ustalenie sposobu prezentacji oznaczonych dokumentów i ich formatów wyjściowych – przygotowanie odpowiednich specyfikacji konwersji formatów i dobór właściwych do tego celu narzędzi (por. punkty ?? i ??).

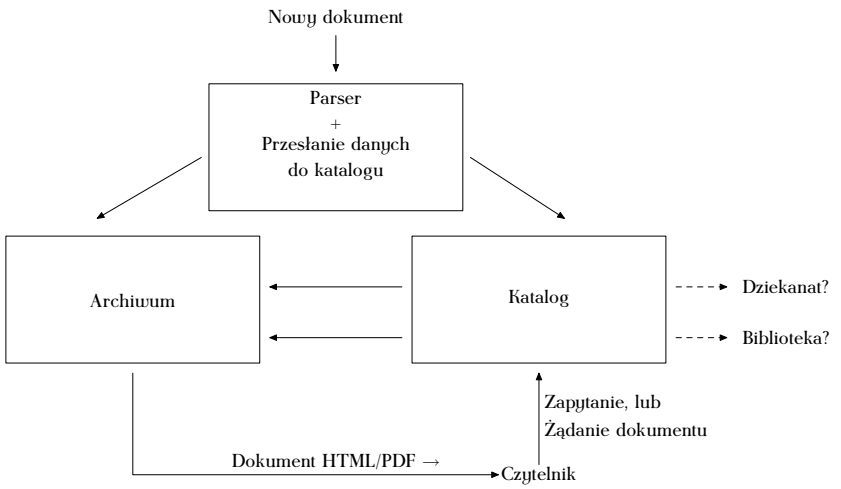
Każdy z wyżej wymienionych problemów stanowi pewne zadanie do wykonania, które może być w dużym stopniu lub całkowicie zautomatyzowane dzięki zastosowaniu odpowiednich narzędzi, co jest jednym z celów i korzyści stosowania systemu opartego na standardzie SGML.

1.2. Sposób wytworzenia zbioru treningowego

Tworzenie DTD [?, ?] powinno zacząć się od analizy posiadanych danych oraz potrzeb dotyczących dokumentów wynikowych i informacji. Rezultatem analizy dokumentów jest wyspecyfikowanie słownika takich elementów oraz, dla każdego typu dokumentu, zależności między tymi elementami. Z tego powodu stworzenie dobrego DTD jest zadaniem niełatwym. Ponadto DTD jest tylko częścią systemu

[1] Definicja typu dokumentu DTD to formalny opis pewnej klasy dokumentów, spis jej elementów składowych (znaczników i ich atrybutów) oraz zasad ich stosowania (hierarchii występowania czy możliwości skracania).

[2] Wybrany edytor winien w maksymalnym stopniu ułatwiać proces tworzenie dokumentów strukturalnych, różniący się od tego, jaki jest stosowany w narzędziach biurowych typu edytor Word.



Rysunek 1.1. Schemat archiwum

Źródło: Opracowanie własne

SGML; oprócz niego potrzebne są narzędzia do formatowania dokumentów, takie jak arkusze stylów DSSSL czy też XSL/XSLT.

Duże koszty implementacji przemawiają często za rozważeniem wykorzystania, czy też adaptacji w systemie SGML udostępnionych publicznie gotowych DTD. W omawianym projekcie postąpiono podobnie, ewentualne opracowanie własnego typu dokumentu odkładając na później. Rozważano wstępnie wykorzystanie trzech publicznie dostępnych typów dokumentów: `etd` zdefiniowanego na potrzeby projektu *Electronic Theses and Dissertations* w Virginia Polytechnic Institute and State University, `TEI` opracowanego w ramach projektu *Text Encoding Iniciative* oraz DocBook opracowany przez konsorcjum firm, głównie sektora IT, używających technologii SGML do tworzenia dokumentacji do swoich produktów.

1.3. Algorytm k-NN

DocBook DTD to typ dokumentu definiujący strukturę i zawartość zbioru znaczników SGML, służących do dokumentacji oprogramowania i różnego rodzaju dokumentacje techniczne [**?**, s. 123]. Posiada ona bardzo rozbudowaną hierarchię

znaczników do budowy struktur książek, dokumentacji technicznej itp dokumentów. DTD dla tego typu dokumentu jest dostępne zarówno w standardzie SGML jak i XML. Dostępna jest także uproszczona wersja DTD o nazwie SimpleDocbook. Zaletą używania wersji uproszczonej jest dużo łatwiejsze posługiwanie się nią, z uwagi na znacznie mniejszą liczbę znaczników, por. tabela 1.1.

Typ dokumentu	Liczba elementów	Liczba encji
Docbook	357	1814
SimpleDocBook	93	234
TEI	brak danych	3
HTML	98	234

Tabela 1.1. Porównanie wielkości popularnych definicji typu dokumentu

Źródło: Obliczenia własne

DocBook jest używany obecnie w większości zarówno komercyjnych jak i niekomercyjnych projektach tworzenia dokumentacji komputerowej (np. dokumentacja dla takich projektów jak: LDP – *Linux Documentation Project*, *RedHat Gnome Desktop*, *Postgres SQL RDBMS*, *PHP3 Hypertext Preprocesor* czy dokumentacja do systemu FreeBSD).

1.4. Metryka miejska

Pierwszym elementem systemu SGML jest edytor strukturalny, zwany też edytorem sterowanym składnią. W tradycyjnych edytorach działających według paradygmatu WYSIWYG autor wprowadzając tekst określa na bieżąco jego wygląd, mając zarówno w jednym jak i drugim przypadku dużą swobodę. Taki sposób pracy w przypadku masowego tworzenia dokumentów jest zupełnie nieefektywny. W szczególności dotyczy to pracy wielu autorów nad jednym lub wieloma dokumentami o zuniformizowanym formacie, czy też tworzenia dokumentów, o których z góry wiadomo, że będą prezentowane w różny sposób.

Pewnym rozwiązaniem opisywanych wyżej problemów jest wykorzystanie szablonów, w które, od jakiegoś czasu są wyposażane wszystkie popularne edytory

biurowe. Posługując się szablonami, autor nie definiuje wyglądu samodzielnie tylko wypełnia treścią zdefiniowany z góry szablon dokumentu. Stanowiąc niewątpliwy krok do przodu takie podejście nie rozwiązuje jednak wszystkich problemów.

Komercyjne edytory SGML, są w chwili obecnej ciągle bardzo drogie, zaś oprogramowanie rozprowadzane na różnego rodzaju licencjach *Open Source* nie oferuje jeszcze wygody pracy, do której przyzwyczaili się użytkownicy edytorów biurowych.

1.5. Jednokierunkowe, wielowarstwowe sieci neuronowe

Jedynym dostępnym w chwili obecnej, efektywnym, tanim (darmowy) i dostępnym na wielu platformach systemowo-sprzętowych środowisku do tworzenia dokumentów strukturalnych jakim jest edytor Emacs z pakietem psgml. Środowisko to wspomaga autora poprzez: kolorowanie składni dokumentów SGML i aplikacji SGML, automatyczne uzupełnianie brakujących znaczników, automatyczną kontrolę jakie w danym kontekście można wstawiać znaczniki i atrybuty, wyświetlanie informacje odnośnie możliwych i domyślnych wartości. Wszystkie funkcje są dostępne za pomocą odpowiednie skrótów klawiszowych a także dostępne poprzez wybór wskaźnikiem myszy odpowiedniej pozycji z menu. Pakiet ten umożliwia również wywołanie zewnętrznego parsera SGML celem weryfikacji poprawności dokumentu.

1.6. Konwolucyjne sieci neuronowe

Jedynym dostępnym w chwili obecnej, efektywnym, tanim (darmowy) i dostępnym na wielu platformach systemowo-sprzętowych środowisku do tworzenia dokumentów strukturalnych jakim jest edytor Emacs z pakietem psgml. Środowisko to wspomaga autora poprzez: kolorowanie składni dokumentów SGML i aplikacji SGML, automatyczne uzupełnianie brakujących znaczników, automatyczną kontrolę jakie w danym kontekście można wstawiać znaczniki i atrybuty, wyświetlanie informacje odnośnie możliwych i domyślnych wartości. Wszystkie funkcje są dostępne za pomocą odpowiednie skrótów klawiszowych a także dostępne poprzez

wybór wskaźnikiem myszy odpowiedniej pozycji z menu. Pakiet ten umożliwia również wywołanie zewnętrznego parsera SGML celem weryfikacji poprawności dokumentu.

ROZDZIAŁ 2

Implementacja aplikacji do rozpoznawania tekstu

Systemy SGML, ze względu na mnogość funkcji jakie spełniają i ich kompleksowe podejście do oznakowywania i przetwarzania dokumentów tekstowych, są bardzo skomplikowane. Możemy wyróżnić dwa podejścia do budowy takich systemów. Z jednej strony, buduje się systemy zindywidualizowane, oparte o specyficzne narzędzia tworzone w takich językach, jak: C, C++, Perl czy Python. Edytory strukturalne, filtry do transformacji formatów czy parsery i biblioteki przydatne do konstrukcji dalszych narzędzi, tworzone są według potrzeb określonych, pojedynczych systemów.

Z drugiej strony, twórcy oprogramowania postanowili pójść krok dalej i połączyć te różne narzędzia w jedną całość. Tą całość miał stanowić DSSSL lub jego XML-owy odpowiednik – standard XSL. Ze względu na oferowane możliwości można twierdzić, że tworzenie i używanie narzędzi implementujących standard DSSSL/XSL, jest najwłaściwszym podejściem. Przemawiają za tym różne argumenty, ale najważniejszym z nich jest to, że mamy tu możliwość stworzenia niezależnego od platformy programowej i narzędziowej zbioru szablonów – przepisów jak przetwarzać dokumenty SGML.

2.1. Xamarin.Android i Xamarin.iOS

DSSSL (*Document Style Semantics and Specification Language*) – to międzynarodowy standard ściśle związany ze standardem SGML. Standard ten, można podzielić na następujące części:

- język transformacji (*transformation language*). To definicja języka służącego

do transformacji dokumentu oznaczonego znacznikami zgodnie z pewnym DTD na dokument oznaczony zgodnie z innym DTD.

- język stylu (*style language*) opisujący sposób formatowania dokumentów SGML.

- język zapytań (*query language*) służy do identyfikowania poszczególnych fragmentów dokumentu SGML.

Opisane główne części składowe standardu DSSSL dają obraz tego, jak wiele aspektów przetwarzania zostało zdefiniowanych i jak skomplikowany jest to problem. Jest to głównym powodem tego, że mimo upływu kilku lat od zdefiniowania standardu nie powstały ani komercyjne ani wolnodostępne aplikacje wspierające go w całości. Istnieją natomiast *nieliczne* narzędzia realizujące DSSSL w ograniczonym zakresie, głównie w części definiującej język stylu, który odpowiada za opatrzenie dokumentu czysto strukturalnego w informacje formatujące. Daje to możliwość publikacji dokumentów SGML zarówno w postaci elektronicznej, hipertekstowej czy też drukowanej.

2.2. Xamarin.Forms

DSSSL (*Document Style Semantics and Specification Language*) – to międzynarodowy standard ściśle związany ze standardem SGML. Standard ten, można podzielić na następujące części:

- język transformacji (*transformation language*). To definicja języka służącego do transformacji dokumentu oznaczonego znacznikami zgodnie z pewnym DTD na dokument oznaczony zgodnie z innym DTD.

- język stylu (*style language*) opisujący sposób formatowania dokumentów SGML.

- język zapytań (*query language*) służy do identyfikowania poszczególnych fragmentów dokumentu SGML.

Opisane główne części składowe standardu DSSSL dają obraz tego, jak wiele aspektów przetwarzania zostało zdefiniowanych i jak skomplikowany jest to problem. Jest to głównym powodem tego, że mimo upływu kilku lat od zdefiniowania standardu nie powstały ani komercyjne ani wolnodostępne aplikacje wspierające go w całości. Istnieją natomiast *nieliczne* narzędzia realizujące DSSSL w ograniczonym zakresie, głównie w części definiującej język stylu, który odpowiada za opatrzenie dokumentu czysto strukturalnego w informacje formatujące. Daje to możliwość publikacji dokumentów SGML zarówno w postaci elektronicznej, hipertekstowej czy też drukowanej.

2.3. Microsoft Computer Vision API

Tak jak XML jest *uproszczoną* wersją standardu SGML, tak XSL jest uproszczonym odpowiednikiem standardu DSSSL. W szczególności, wyróżnić można w tym standardzie następujące części składowe:

- język transformacji (XSLT) To definicja języka służącego do transformacji dokumentu.
- język zapytań (XPath) służy do identyfikowania poszczególnych fragmentów dokumentu.
- język stylu deefiniujący sposób formatowania dokumentów XML.

2.4. Tesseract API

W celu wykorzystania standardu SGML do przetwarzania dokumentów, niezbędne jest zebranie odpowiedniego zestawu narzędzi. Narzędzi do przetwarzania dokumentów SGML jest wiele. Są to zarówno całe systemy zintegrowane, jak i poszczególne programy, biblioteki czy skrypty wspomagające.

2.5. Microsoft Azure for Machine Learning

Program `nsgmls` (z pakietu `SP` Jamesa Clarka) jest doskonałym parserem dokumentów SGML, dostępnym publicznie. Parser `nsgmls` jest dostępny w postaci źródłowej oraz w postaci programów wykonywalnych przygotowanych na platformę MS Windows, Linux/Unix i inne. Oprócz analizy poprawności dokumentu parser ten umożliwia również konwersję danych do formatu ESIS, który wykorzystywany jest jako dane wejściowe przez wiele narzędzi do przetwarzania i formatowania dokumentów SGML. Dodatkowymi, bardzo przydatnymi elementami pakietu `SP` są: program `sgmlnorm` do normalizacji, program `sx` służący do konwersji dokumentu SGML na XML oraz biblioteki programistyczne, przydatne przy tworzeniu specjalistycznych aplikacji służących do przetwarzania dokumentów SGML.

W przypadku dokumentów XML publicznie dostępnych, parserów jest w chwili obecnej kilkadziesiąt. Do popularniejszych zaliczyć można Microsoft Java XML Parser firmy Microsoft, LT XML firmy Language Technology Group, Exapt oraz XP (James Clark)

2.6. Tensorflow

Do tej kategorii oprogramowania zaliczamy przeglądarki dokumentów SGML oraz serwery sieciowe wspomagające standard SGML, przy czym rozwiązań wspierających standard XML jest już w chwili obecnej dużo więcej i są dużo powszechniejsze.

Jeżeli chodzi o przeglądarki to zarówno Internet Explorer jak i Netscape umożliwiają bezpośrednie wyświetlenie dokumentów XML; ponieważ jednak nie wspierają w całości standardu XML, prowadzi to ciągle do wielu problemów[1].

[1] Z innych mniej popularnych rozwiązań można wymienić takie aplikacje, jak: HyBrick SGML Browser firmy Fujitsu Limited, Panorama Publisher firmy InterLeaf Inc, DynaText firmy Inso Corporation czy darmowy QWeb. W przypadku serwerów zwykle dokonują one transformacji „w locie" żądanych dokumentów na format HTML (rzadziej bezpośrednio wyświetlają dokumenty XML). Ta kategoria oprogramowania ma, z punktu widzenia projektu, znaczenie drugorzędne.

ROZDZIAŁ 3

Metryki oraz testy

Stosując wersję XML typu DocBook można wykorzystać szablony stylów przygotowane w standardzie XSL (autor N. Walsh). W chwili obecnej są dostępne narzędzia umożliwiające przetworzenie dokumentów XML do postaci drukowanej (Adobe PDF) oraz hipertekstowej (HTML).

Podobnie jak w przypadku szablonów DSSSL, szablony stylów XSL są sparametryzowane i udokumentowane i dzięki temu łatwe w adaptacji. Do zamiany dokumentu XML na postać prezentacyjną można wykorzystać jeden z dostępnych publicznie procesorów XSLT (por. tabela 3.1).

3.1. Testy wydajnościowe

Stosując wersję XML typu DocBook można wykorzystać szablony stylów przygotowane w standardzie XSL (autor N. Walsh). W chwili obecnej są dostępne narzędzia umożliwiające przetworzenie dokumentów XML do postaci drukowanej (Adobe PDF) oraz hipertekstowej (HTML).

3.2. Testy zgodności

Stosując wersję XML typu DocBook można wykorzystać szablony stylów przygotowane w standardzie XSL (autor N. Walsh). W chwili obecnej są dostępne narzędzia umożliwiające przetworzenie dokumentów XML do postaci drukowanej (Adobe PDF) oraz hipertekstowej (HTML).

3.3. Testy użyteczności

Stosując wersję XML typu DocBook można wykorzystać szablony stylów przygotowane w standardzie XSL (autor N. Walsh). W chwili obecnej są dostępne narzędzia umożliwiające przetworzenie dokumentów XML do postaci drukowanej (Adobe PDF) oraz hipertekstowej (HTML).

3.4. Cross Validation

Stosując wersję XML typu DocBook można wykorzystać szablony stylów przygotowane w standardzie XSL (autor N. Walsh). W chwili obecnej są dostępne narzędzia umożliwiające przetworzenie dokumentów XML do postaci drukowanej (Adobe PDF) oraz hipertekstowej (HTML).

3.5. Macierze błedu

Stosując wersję XML typu DocBook można wykorzystać szablony stylów przygotowane w standardzie XSL (autor N. Walsh). W chwili obecnej są dostępne narzędzia umożliwiające przetworzenie dokumentów XML do postaci drukowanej (Adobe PDF) oraz hipertekstowej (HTML).

3.6. Metryki wyliczane z kodu źródłowego

Stosując wersję XML typu DocBook można wykorzystać szablony stylów przygotowane w standardzie XSL (autor N. Walsh). W chwili obecnej są dostępne narzędzia umożliwiające przetworzenie dokumentów XML do postaci drukowanej (Adobe PDF) oraz hipertekstowej (HTML).

3.7. Macierze wyliczane z diagramów

Stosując wersję XML typu DocBook można wykorzystać szablony stylów przygotowane w standardzie XSL (autor N. Walsh). W chwili obecnej są dostępne narzędzia

umożliwiające przetworzenie dokumentów XML do postaci drukowanej (Adobe PDF) oraz hipertekstowej (HTML).

3.8. Macierze pomiaru wspólnego kodu

Stosując wersję XML typu DocBook można wykorzystać szablony stylów przygotowane w standardzie XSL (autor N. Walsh). W chwili obecnej są dostępne narzędzia umożliwiające przetworzenie dokumentów XML do postaci drukowanej (Adobe PDF) oraz hipertekstowej (HTML).

Nazwa	Autor	Adres URL
`sablotron`	Ginger Alliance	`http://www.gingerall.com`
`Xt`	J. Clark	`http://www.jclark.com`
`4XSLT`	FourThought	`http://www.fourthought.com`
`Saxon`	Michael Kay	`http://users.iclway.co.uk/mhkay/saxon`
`Xalan`	Apache XML Project	`http://xml.apache.org`

Tabela 3.1. Publicznie dostępne procesory XLST

Źródło: Opracowanie własne

XSL:FO jest skomplikowanym językiem o dużych możliwościach, zawierającym ponad 50 różnych „obiektów formatujących", począwszy od najprostszych, takich jak prostokątne bloki tekstu poprzez wyliczenia, tabele i odsyłacze. Obiekty te można formatować wykorzystując przeszło 200 różnych właściwości (*properties*), takich jak: kroje, odmiany i wielkości pisma, odstępy, kolory itp. W tym dokumencie przedstawione jest absolutne miniumum informacji na temat standardu XSL:FO.

Cały dokument XSL:FO zawarty jest wewnątrz elementu `fo:root`. Element ten zawiera (w podanej niżej kolejności):

- dokładnie jeden element `fo:layout-master-set` zawierający szablony określające wygląd poszczególnych stron oraz sekwencji stron (te ostatnie są opcjonalne, ale typowo są definiowane);

- zero lub więcej elementów `fo:declarations`;
- jeden lub więcej elementów `fo:page-sequance` zawierających treść formatowanego dokumentu wraz z opisem jego sformatowania i podziału na strony.

ROZDZIAŁ 4

Podsumowanie i wnioski

Stosując wersję XML typu DocBook można wykorzystać szablony stylów przygotowane w standardzie XSL (autor N. Walsh). W chwili obecnej są dostępne narzędzia umożliwiające przetworzenie dokumentów XML do postaci drukowanej (Adobe PDF) oraz hipertekstowej (HTML).

Podobnie jak w przypadku szablonów DSSSL, szablony stylów XSL są sparametryzowane i udokumentowane i dzięki temu łatwe w adaptacji. Do zamiany dokumentu XML na postać prezentacyjną można wykorzystać jeden z dostępnych publicznie procesorów XSLT (por. tabela 3.1).

4.1. Wady oraz zalety aplikacji wieloplatformowych

Stosując wersję XML typu DocBook można wykorzystać szablony stylów przygotowane w standardzie XSL (autor N. Walsh). W chwili obecnej są dostępne narzędzia umożliwiające przetworzenie dokumentów XML do postaci drukowanej (Adobe PDF) oraz hipertekstowej (HTML).

4.2. Uczenie maszynowe w aplikacjach mobilnych

Stosując wersję XML typu DocBook można wykorzystać szablony stylów przygotowane w standardzie XSL (autor N. Walsh). W chwili obecnej są dostępne narzędzia umożliwiające przetworzenie dokumentów XML do postaci drukowanej (Adobe PDF) oraz hipertekstowej (HTML).

4.3. Koszt

Stosując wersję XML typu DocBook można wykorzystać szablony stylów przygotowane w standardzie XSL (autor N. Walsh). W chwili obecnej są dostępne narzędzia umożliwiające przetworzenie dokumentów XML do postaci drukowanej (Adobe PDF) oraz hipertekstowej (HTML).

Zakończenie

Możliwości, jakie stoją przed archiwum prac magisterskich opartych na XML-u, są ograniczone jedynie czasem, jaki należy poświęcić na pełną implementację systemu. Nie ma przeszkód technologicznych do stworzenia co najmniej równie doskonałego repozytorium, jak ma to miejsce w przypadku ETD. Jeżeli chcemy w pełni uczestniczyć w rozwoju nowej ery informacji, musimy szczególną uwagę przykładać do odpowiedniej klasyfikacji i archiwizacji danych. Sądzę, że język XML znacznie to upraszcza.

Oświadczenie

Ja, niżej podpisany(a) oświadczam, iż przedłożona praca dyplomowa została wykonana przeze mnie samodzielnie, nie narusza praw autorskich, interesów prawnych i materialnych innych osób.

.....	.....
data	podpis